

1995

COSSAP simulation model of DS-CDMA indoor microwave ATM LAN

Keng T. Tan
Edith Cowan University

Follow this and additional works at: https://ro.ecu.edu.au/theses_hons



Part of the [Signal Processing Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

Tan, K. T. (1995). *COSSAP simulation model of DS-CDMA indoor microwave ATM LAN*.
https://ro.ecu.edu.au/theses_hons/284

This Thesis is posted at Research Online.
https://ro.ecu.edu.au/theses_hons/284

**DEPARTMENT OF COMPUTER AND COMMUNICATION
ENGINEERING
EDITH COWAN UNIVERSITY
JOONDALUP DRIVE
JOONDALUP 6027
WESTERN AUSTRALIA**

***FOURTH YEAR COMPUTER SYSTEM ENGINEERING PROJECT THESIS
ENS4241***

COSSAP Simulation Model of DS-CDMA Indoor Microwave ATM LAN

**by
Keng T. Tan
0926011**

Supervisor: Dr Tadeusz Wysocki

October 1995

Edith Cowan University

Copyright Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorize you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.
- A reproduction of material that is protected by copyright may be a copyright infringement. Where the reproduction of such material is done without attribution of authorship, with false attribution of authorship or the authorship is treated in a derogatory manner, this may be a breach of the author's moral rights contained in Part IX of the Copyright Act 1968 (Cth).
- Courts have the power to impose a wide range of civil and criminal sanctions for infringement of copyright, infringement of moral rights and other offences under the Copyright Act 1968 (Cth). Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

USE OF THESIS

The Use of Thesis statement is not included in this version of the thesis.

TABLE OF CONTENTS

| | | |
|---------|---|----|
| 1 | Introduction to Spread Spectrum Communication | 13 |
| 1.1 | Introduction | 13 |
| 1.2 | Direct Sequence - Code Division Multiple Access | 13 |
| 1.2.1 | Spread Spectrum | 13 |
| 1.2.2 | Direct Sequence | 16 |
| 1.2.3 | Code Division Multiple Access | 17 |
| 1.3 | Pseudorandom Noise generation | 18 |
| 1.4 | Constraints and limitations of DS-CDMA | 20 |
| 1.5 | Advantages of using DS-CDMA | 23 |
| 1.6 | Low Correlation Functions | 26 |
| 1.6.1 | Orthogonal functions | 27 |
| 1.6.2 | Rademacher-Walsh functions | 28 |
| 1.6.3 | Generation of orthogonal Rademacher-Walsh functions | 32 |
| 1.7 | Indoor Microwave Channel Characteristics | 36 |
| 1.8 | ATM Systems | 39 |
| 1.9 | Channel Modulation Techniques | 39 |
| 1.9.1 | Phase Shift Keying | 40 |
| 1.9.2 | Differential Phase Shift Keying | 42 |
| 1.10 | Proposed System Topology and Overview | 47 |
| 2 | Overview of Communication System Simulation and Analysis Package | 50 |
| 2.1 | Introduction | 50 |
| 2.2 | Communication System Simulation and Analysis Package | 50 |
| 2.3 | Building models in COSSAP | 51 |
| 2.4 | Errors and debugging COSSAP programs | 52 |
| 2.5 | System constraints of COSSAP | 53 |
| 2.6 | Hardware Platform | 55 |
| 3 | COSSAP Implementation of a DS-CDMA transmission network | 56 |
| 3.1 | Introduction | 56 |
| 3.2 | System Model | 56 |
| 3.2.1 | Transmitter Modelling | 58 |
| 3.2.2 | Channel Modelling | 61 |
| 3.2.3 | Receiver Modelling | 62 |
| 3.3 | Functional blocks used in the design and development of the COSSAP simulation model of DS-CDMA indoor microwave ATM LAN | 65 |
| 3.3.1 | PN spectrum spreading and despreading blocks | 66 |
| 3.3.1.1 | Barker sequence spreading and despreading for real signals | 67 |
| 3.3.2 | Transmission channel modulator and demodulator blocks | 67 |
| 3.3.2.1 | QPSK with Trellis Coded Modulation and Viterbi soft decoding | 68 |
| 3.3.2.2 | DQPSK with Trellis Coded Modulation and Viterbi soft decoding | 70 |
| 3.3.2.3 | BPSK with Trellis Coded Modulation and Viterbi soft decoding | 71 |
| 3.3.2.4 | DBPSK with Trellis Coded Modulation and Viterbi soft decoding | 72 |
| 3.3.3 | CDMA multiplexer and demultiplexer blocks | 73 |
| 3.3.3.1 | Walsh sequence code multiplexer and demultiplexer | 73 |
| 3.3.4 | Arithmetic signals adder blocks | 75 |
| 3.4 | Simulation technique for system BER evaluation | 75 |

| | | |
|--|--|-----|
| 4 | Performance evaluation of different CDMA code sequences..... | 77 |
| 4.1 | Introduction | 77 |
| 4.2 | Simulation System Design | 78 |
| 4.2.1 | Simulating varying distance transmitter | 80 |
| 4.3 | CDMA coding..... | 81 |
| 4.3.1 | Optimum Gold code set | 81 |
| 4.3.2 | Rademacher-Walsh orthogonal code sequences | 83 |
| 4.4 | Simulation method | 86 |
| 4.5 | Simulation results..... | 87 |
| 5 | Conclusion | 90 |
| 6 | References | 92 |
| Appendix A: Source, Sink and Matched Filter functional blocks | | 95 |
| A.1 | Real Signal Source Generator (PPGENR) | 96 |
| A.2 | BER Counter..... | 98 |
| A.3 | 0.5 Raised-Cosine Roll-off Matched Filter (rcrfqc) | 101 |
| Appendix B: Pseudorandom Noise spreading and despreading functional blocks | | 103 |
| B.1 | 11-bits Barker Sequence Spreading Functional Block (Barker_sp) | 103 |
| B.2 | 11-bits Barker Sequence Despreading Functional Block (Barker_dsp) | 105 |
| Appendix C: Transmission channel modulator and demodulator blocks | | 107 |
| C.1 | BPSK Modulator (BPSK_T) | 107 |
| C.2 | BPSK Demodulator (BPSK_V)..... | 109 |
| C.3 | DBPSK Modulator (DBPSK_T)..... | 112 |
| C.4 | DBPSK Demodulator (DBPSK_V)..... | 113 |
| C.5 | QPSK Modulator (QPSK_T)..... | 115 |
| C.6 | QPSK Demodulator (QPSK_V) | 116 |
| C.7 | DQPSK Modulator (DQPSK_T) | 118 |
| C.8 | DQPSK Demodulator (DQPSK_V)..... | 119 |
| Appendix D: CDMA multiplexer and demultiplexer blocks..... | | 121 |
| D.1 | 16-bits Rademacher-Walsh Orthogonal Sequence Multiplexer Functional Block (Walsh_Mux) | 121 |
| D.1.1 | 16-bits Rademacher-Walsh Orthogonal Sequence Multiplexer Functional Block (Walsh_Mux_QC) - Complex Signals | 123 |
| D.2 | 16-bits Rademacher-Walsh Orthogonal Sequence Demultiplexer Functional Block (Walsh_DMux) | 125 |
| D.2.1 | 16-bits Rademacher-Walsh Orthogonal Sequence Demultiplexer Functional Block (Walsh_DMux_QC) - Complex Signals | 127 |
| Appendix E: Channel Modelling | | 130 |
| E.1 | Additive White Gaussian Noise (AWGNQC) Channel | 130 |
| E.2 | Rayleigh Distribution Multipath Fading (M_RAYLEIGH) Channel | 132 |
| E.3 | Rician Distribution Multipath Fading (RICE) Channel..... | 134 |
| Appendix F: Arithmetic signals adder blocks..... | | 136 |
| F.1 | Arithmetic Adder for 9 Complex Signals (ADD9_QC) | 136 |
| F.2 | Arithmetic Adder for 16 Complex Signals (ADD16_QC) | 138 |

LIST OF FIGURES

| | | |
|------------|---|-----|
| Figure 1: | Illustration of different multiple-access spread spectrum system [13]. | 18 |
| Figure 2: | PN binary sequence generation using shift registers with feedback. ($n = 4$) | 20 |
| Figure 3: | Illustrations of spread spectrum modulation and demodulation, taking into account the effects of channel noise [13]. | 25 |
| Figure 4: | A set of Rademacher functions. | 29 |
| Figure 5: | Diagram of generator using NAND gates and JK flip-flops [20]. | 33 |
| Figure 6: | Master-slave equivalent circuit [20]. | 34 |
| Figure 7: | Orthogonal Walsh functions representation in waveform and equivalent bits. | 35 |
| Figure 8: | Orthogonal Walsh function representations. (Continue from Figure 7). ... | 36 |
| Figure 9: | Illustration of two ray paths between a transmitter and a receiver resulting in different propagation delays [15]. | 38 |
| Figure 10: | QPSK signal constellations (permitted values of the complex envelope). | 42 |
| Figure 11: | DPSK signal generation [14]. | 43 |
| Figure 12: | DPSK detector. | 46 |
| Figure 13: | Overall system topology. | 48 |
| Figure 14: | Pilot channel | 48 |
| Figure 15: | Access channels. | 49 |
| Figure 16: | A simplified block diagram of the simulated system | 57 |
| Figure 17: | A block diagram transmitter channel representation | 59 |
| Figure 18: | Initial transmitter design in COSSAP | 59 |
| Figure 19: | Rician fading channel with signal power control | 62 |
| Figure 20: | A block diagram representation of a receiver channel with noise | 63 |
| Figure 21: | Initial receiver design in COSSAP | 63 |
| Figure 22: | COSSAP block of Walsh_DMux. | 64 |
| Figure 23: | Trellis state diagram for QPSK. | 69 |
| Figure 24: | QPSK constellation mapping. | 70 |
| Figure 25: | Trellis state diagram for BPSK. | 71 |
| Figure 26: | BPSK constellation mapping. | 72 |
| Figure 27: | Block diagram of simulated DS-CDMA system | 78 |
| Figure 28: | Cross-correlation of the 7-bits Gold code sequences. | 82 |
| Figure 29: | Cross-correlation of the 16-bits orthogonal Rademacher-Walsh code sequences | 84 |
| Figure 30: | COSSAP model of 9 channels DS-CDMA system simulation. | 86 |
| Figure 31: | BER of the DS-CDMA system using 7-bits Gold code sequences | 87 |
| Figure 32: | BER of DS-CDMA system using the 16-bits Rademacher-Walsh sequences | 88 |
| Figure 33: | COSSAP model (ppgenr). | 97 |
| Figure 34: | COSSAP model of the BER counter | 100 |
| Figure 35: | COSSAP model (rcrfqc) | 102 |

| | | |
|------------|------------------------------------|-----|
| Figure 36: | COSSAP model (Barker_sp) | 104 |
| Figure 37: | COSSAP model (Barker_dsp) | 106 |
| Figure 38: | COSSAP model (BPSK_T) | 108 |
| Figure 39: | COSSAP model (BPSK_V) | 110 |
| Figure 40: | COSSAP model (DBPSK_T) | 112 |
| Figure 41: | COSSAP model (DBPSK_V) | 113 |
| Figure 42: | COSSAP model (QPSK_T) | 115 |
| Figure 43: | COSSAP model (QPSK_V) | 116 |
| Figure 44: | COSSAP model (DQPSK_T) | 118 |
| Figure 45: | COSSAP model (DQPSK_V) | 119 |
| Figure 46: | COSSAP model (Walsh_Mux) | 122 |
| Figure 47: | COSSAP model (Walsh_Mux_QC) | 124 |
| Figure 48: | COSSAP model (Walsh_DMux) | 126 |
| Figure 49: | COSSAP model (Walsh_DMux_QC) | 128 |
| Figure 50: | COSSAP model (AWGNQC) | 131 |
| Figure 51: | COSSAP model (M_RAYLEIGH) | 133 |
| Figure 52: | COSSAP model (RICE) | 135 |
| Figure 53: | COSSAP model (ADD9_QC) | 137 |
| Figure 54: | COSSAP model (ADD16_QC) | 139 |

LIST OF TABLES

| | | |
|-----------|---|-----|
| Table 1: | The bit streams $b'(t)$ represents the message to be transmitted [23]. | 44 |
| Table 2: | Logical operation of logic circuit. | 45 |
| Table 3: | Constraints of COSSAP SDS | 53 |
| Table 4: | Constraints of COSSAP model libraries | 53 |
| Table 5: | Constraints of COSSAP .mdef file | 54 |
| Table 6: | Constraints of COSSAP CONFIGURATOR Tool | 54 |
| Table 7: | Constraints of COSSAP CHART Tool | 54 |
| Table 8: | Constraints of COSSAP REPORT Tool | 55 |
| Table 9: | Initial system specifications for ATM WLAN | 56 |
| Table 10: | User specific CDMA 7-bits Gold code set | 81 |
| Table 11: | Cross-correlation between the Gold code sequences | 83 |
| Table 12: | User specific CDMA orthogonal 16-bits Rademacher-Walsh code set | 84 |
| Table 13: | Cross-correlation between the 16-bits Rademacher-Walsh code sequences | 85 |
| Table 14: | Configuration file for COSSAP model ppgenr | 96 |
| Table 15: | A sample of input data file for ppgenr | 97 |
| Table 16: | Configuration file for the COSSAP model of BER Counter | 98 |
| Table 17: | Configuration file for COSSAP model rcrfqc | 101 |
| Table 18: | Configuration file for COSSAP model Barker_sp | 103 |
| Table 19: | Configuration file for COSSAP model Barker_dsp | 105 |
| Table 20: | 11-bits Barker sequence input data file | 106 |
| Table 21: | Configuration file for COSSAP model BPSK_T | 107 |
| Table 22: | Configuration file for COSSAP model BPSK_V | 109 |
| Table 23: | Trellis/Viterbi encoding/decoding parameter file for COSSAP model BPSK_T & BPSK_V | 111 |
| Table 24: | Configuration file for COSSAP model DBPSK_T | 112 |
| Table 25: | Configuration file for COSSAP model DBPSK_V | 113 |
| Table 26: | Trellis/Viterbi encoding/decoding parameter file for COSSAP model DBPSK_T & DBPSK_V | 114 |
| Table 27: | Configuration file for COSSAP model QPSK_T | 115 |
| Table 28: | Configuration file for COSSAP model QPSK_V | 116 |
| Table 29: | Trellis/Viterbi encoding/decoding parameter file for COSSAP model QPSK_T & QPSK_V | 117 |
| Table 30: | Configuration file for COSSAP model DQPSK_T | 118 |
| Table 31: | Configuration file for COSSAP model DQPSK_V | 119 |
| Table 32: | Trellis/Viterbi encoding/decoding parameter file for COSSAP model DQPSK_T & DQPSK_V | 120 |
| Table 33: | Configuration file for COSSAP model Walsh_Mux | 121 |
| Table 34: | Configuration file for COSSAP model Walsh_Mux_QC | 123 |
| Table 35: | Configuration file for COSSAP model Walsh_DMux | 125 |

| | | |
|-----------|---|-----|
| Table 36: | Configuration file for COSSAP model Walsh_DMux_QC | 127 |
| Table 37: | Sample of 16-bits Rademacher-Walsh sequence input data file | 129 |
| Table 38: | Configuration file for COSSAP model AWGNQC | 130 |
| Table 39: | Configuration file for COSSAP model M_RAYLEIGH | 132 |
| Table 40: | Configuration file for COSSAP model RICE..... | 134 |
| Table 41: | Configuration file for COSSAP model ADD9_QC..... | 136 |
| Table 42: | Configuration file for COSSAP model ADD16_QC..... | 138 |

6, Park Close

GREENWOOD WA 6024

October 30, 1995

Chairperson of Department

Department of Computer and Communication Engineering

Edith Cowan University

Joondalup Drive

JOONDALUP WA 6027

Dear Sir

This thesis entitled *COSSAP Simulation Model of DS-CDMA Indoor Microwave ATM LAN* is presented as part of the requirements for the award of the degree for the Bachelor of Engineering in Computer Science (Hons.). I certify that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text.

I Acknowledgments

I take this opportunity to thank many who have contributed to the production of this thesis. Of these, the following individuals deserve special mention.

Dr Tadeusz Wysocki for his supervision, tremendous assistance and encouragements.

Without him I would not have been able to be involved with such a challenging project nor to work within the *Cooperative Research Centre for Broadband Telecommunication and Networking*. Most of all, I am forever in debt to his guidance and concerns for me both as a mentor and a friend.

Dr Hans J. Zepernick for bringing a new perspective to my outlook at my work.

My beloved parents, *Nam Tat Tan and Kwai Ying Lau*, for their everlasting love, devotion and investments in my education and in me.

My beloved brother, *Kuei Chiao Tan*, who is always my best friend and dearest brother.

Fellow research team members of the research team, *Jason Hislop, Ted Walker and Beata Wysocki* for their ever present support and encouragement within our growing research group.

Professors, lecturers and academic friends within the School and Department of Engineering at Edith Cowan University, Joondalup for all their guidance and generous assistance.

Last but never the least, all my wonderful non-academic friends, the *MacGregors, Hogans* and all my other friends, for their forever lasting friendship and concern.

II Extended Abstract

The possibilities of using spread spectrum techniques in indoor radiowave communication have gained a lot of research interest. International communications regulation bodies like the Institute of Electrical and Electronic Engineers (IEEE) and the International Telecommunication Union (ITU) have established recommendation committees and set aside radio spectrum bands for these anticipated developments [1,2]. Many low speed communication networks using spread spectrum techniques have been successfully developed recently [3,4]. Developments in the higher speed and asynchronous spread spectrum transmission networks are still to follow [5]. The lag in development progress in these type of systems is partly due to the difficulties in testing and simulating designs for asynchronous high speed spread spectrum systems. Many communication design packages requires time synchronisation rather than be event driven. For those few testbed models that exist, they tend to cater for a specific implementation rather than be generic enough to cross evaluate different spread spectrum implementations [6,7].

This thesis presents an original work in the area of designing and implementing a simulation testbed for modelling a high speed spread spectrum Asynchronous Transfer Mode (ATM) Local Area Network (LAN). The spread spectrum technique used in this LAN model is Direct Sequence Code Division Multiple Access (DS-CDMA). The simulation model includes at least a physical layer of such a LAN, embedded into the COSSAP¹ simulation environment, and has been fully tested. All the newly developed building blocks are comprised of standard blocks from the COSSAP libraries or compatible user-built primitive blocks (only where it is

1. Communication System Simulation and Analysis Package (Ver.6.7) by Synopsys.

absolutely necessary), and are flexible enough to allow the modification of simulation or model parameters; such as the number of signal channels, modulation method used, different spreading code sequences and so on. All these changes can be made with minimal effort.

Another significant contribution made in this thesis is the extended research into evaluating the Bit Error Rate (BER) performance of different spread spectrum CDMA coding schemes for an indoor microwave ATM LAN [8]. Different spread spectrum CDMA coding schemes are compared for their transmission error rate in Additive White Gaussian Noise (AWGN) channel with varying transmitted signal power and at different channel Signal to Noise Ratio (SNR) levels. Since a wireless microwave channel is very prone to transmission errors, a major contribution of the simulation testbed developed in this thesis is its use in the finding of an optimal physical layer transmission scheme with the best Bit Error Rate (BER) performance in an indoor environment.

The layout of this thesis is as follows:

- Chapter 1 begins this thesis by presenting the theoretical foundation used in the development of the simulation model of the DS-CDMA indoor microwave ATM LAN. All the system components pertinent to the development of the simulation model are discussed. Lengthy discussion into the orthogonal CDMA coding scheme initially investigated is also given. Finally a preliminary overall system layout architecture is proposed.
- Chapter 2 gives an introduction to the software simulation environment used - COSSAP. The use and constraints of COSSAP are illustrated and explained within the context of using it to develop the simulation model of a DS-CDMA indoor microwave ATM LAN.

-
- Chapter 3 introduces the developed simulation model of DS-CDMA indoor microwave ATM LAN. Each functional block of the developed model is highlighted and explained. All COSSAP models developed are also included in the Appendixes.
 - Chapter 4 looks at the important use of the simulation model in evaluating the BER performance between different CDMA coding schemes.
 - Chapter 5 concludes the thesis with significant areas for further research based on the work presented in this thesis.

1 Introduction to Spread Spectrum Communication

1.1 Introduction

This chapter gives the theoretical background for an understanding, within the context of the simulation, of multiple access spread spectrum communication, especially the DS-CDMA.

Included are: aspects of its generation, limitations and advantages. An overview of the properties of both the Pseudorandom Noise (PN) and the CDMA multiplexing codes, such as low correlation code sequences and orthogonal functions are presented with an introduction to a specific orthogonal code series - the Rademacher-Walsh functions. Some characteristics of a microwave channel, a brief description of ATM systems, and possible channel modulation techniques are also discussed. Finally, this chapter is concluded with a preliminary proposal for an overall system architecture.

1.2 Direct Sequence - Code Division Multiple Access

1.2.1 Spread Spectrum

CDMA is a form of spread spectrum communication. A definition of spread spectrum communication is as follows [10]:

“Spread spectrum is a mean of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by means of a code which is independent of the data, and a synchronised reception with the code at the receiver is used for despreading and subsequent data recovery.”

The basis for spread spectrum technology is also illustrated by C. E. Shannon in the form of channel capacity [11] given for the additive white Gaussian noise channel:

$$C = B \log_2 \left(1 + \frac{S}{N} \right) \quad (1)$$

where C - channel capacity in bits per second (bps),

B - bandwidth in Hertz,

N - noise power,

S - signal power.

Letting C to be the desired system information rate and changing the bases of the logarithmic function, we arrive at

$$\frac{C}{B} = 1.44 \ln \left(1 + \frac{S}{N} \right) \quad (2)$$

and, for small $\frac{S}{N}$, say $\frac{S}{N} \leq 0.1$ (as it is to be in an anti-jam system), one can use only the first term of a $\ln \left(1 + \frac{S}{N} \right)$ series expansion [11]:

$$\ln \left(1 + \frac{S}{N} \right) = \frac{S}{N} - \frac{1}{2} \left(\frac{S}{N} \right)^2 + \frac{1}{3} \left(\frac{S}{N} \right)^3 - \frac{1}{4} \left(\frac{S}{N} \right)^4 \dots \left(-1 < \frac{S}{N} < 1 \right)$$

Therefore, formula (2) simplifies to:

$$\frac{C}{B} = 1.44 \frac{S}{N} \quad (3)$$

From formula (3) we find

$$\frac{N}{S} = 1.44 \frac{B}{C} \approx \frac{B}{C} \quad (4)$$

Hence,

$$B \approx \frac{NC}{S} \quad (5)$$

And because the information-error rate, R_e , is inversely proportional to the channel capacity, C :

$$R_e \propto \frac{1}{C} \quad (6)$$

Thus from formula (5), for any given signal-to-noise ratio,

$$B \propto C \quad (7)$$

With formula (6) and (7), we obtain

$$R_e \propto \frac{1}{B} \quad (8)$$

Thus we see that for any given signal-to-noise ratio we can lower the information-error rate by increasing the bandwidth used to transfer the information.

1.2.2 Direct Sequence

DS is the method where the spectrum is spread by a fast, pseudorandomly generated sequence (i.e. a large number of coded bits called *chips*), which causes phase transitions in the carrier containing the data [10]. Other methods of spectrum spreading are “frequency hopping”, in which the carrier is caused to shift frequency in a pseudorandom or fixed manner, and “time hopping”, where bursts of signals are initiated at pseudorandom times.

Example 1:

For example, if an information bit rate $R = 2$ Mbps (minimum transmission bit rate for the proposed ATM LAN) is used, it needs an information bandwidth $B \approx 2$ MHz. If each bit of 2 Mbps is coded by 40 chips, then the chip rate is 80 Mbps, which needs a DS bandwidth $B_{ss} = 80$ MHz (the allocated bandwidth for microwave LANs). The bandwidth is thus spreading from 2 MHz to 80 MHz. The spectrum spreading in DS is measured by the processing gain (PG) in dB [11]

$$PG = 10 \log \frac{B_{ss}}{B} \quad (9)$$

The PG in this example is therefore equal to 16.02 dB.

Processing gain is very important as it has a strong influence on system error and interference performance. In fact, from formula (9) it follows that

$$10^{\frac{PG}{10}} \propto \frac{1}{P_e} \quad (10)$$

where

P_e - system's probability of error; ($P_e \propto R_e$)

In Example 1, with a PG = 16.02 dB, the jamming power (interference) is effectively suppressed by 16.02 dB. This means with a PG = 16.02 dB, our DS spread spectrum system can tolerate a 16.02 dB greater jammer power than a similar system without direct sequence spectrum spreading, at the same level of error probability [12].

1.2.3 Code Division Multiple Access

Frequency Division Multiple Access (FDMA) systems allocate a different narrowband frequency slot to each different access signal. Time Division Multiple Access (TDMA) systems separate different multiple access signals by transmitting each signal at different time slots. Different access signals in different time slots are kept apart from one another using guard time gaps. Unlike FDMA or TDMA, CDMA systems use low correlation codes to separate one signal from another. Hence with CDMA, all signals share the same wideband channel. The difference between CDMA, FDMA and TDMA is illustrated in Figure 1.

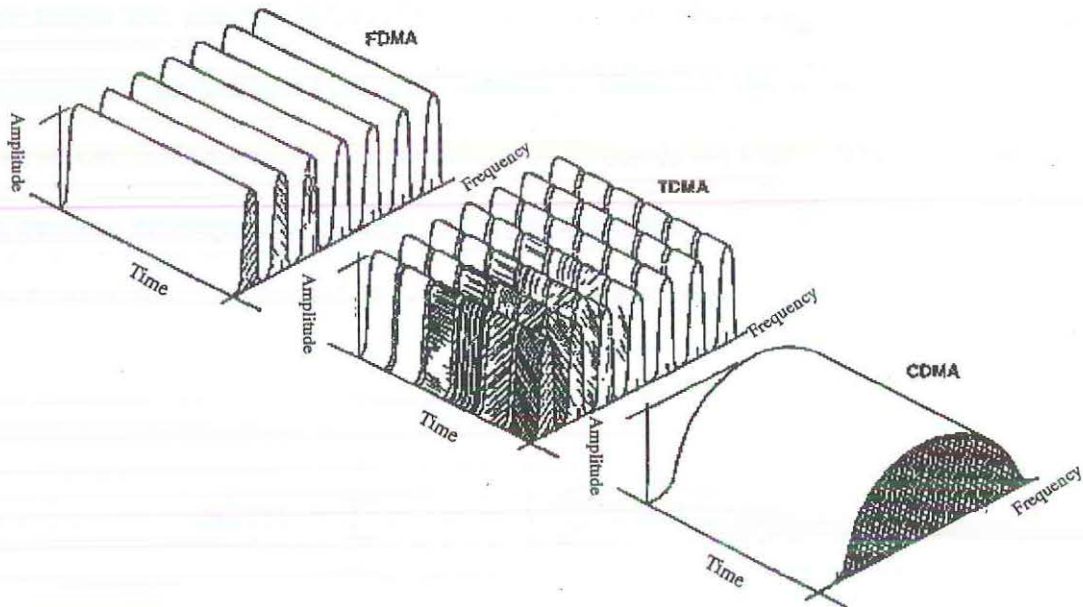


Figure 1: Illustration of different multiple-access spread spectrum system [13].

1.3 Pseudorandom Noise generation

To spread a narrowband signal to a wideband spectrum, we multiply a binary PN sequence with this narrowband signal. For this purpose, the required PN sequence is defined to be a maximum-length linear recurring sequence of modulo-2 (i.e. a binary sequence). For a binary sequence, the following properties are associated with randomness [14]:

- A balance of 1 and 0 terms. (More precisely, the disparity [14] is not to exceed 1.

Thus $\left| \sum_{n=1}^p (a_n - 0.5) \right| \leq 0.5$ i.e. for a PN binary sequence the numbers of the PN code differs only by one.)

-
- Two sequences of consecutive-like elements of length n for each sequence of consecutive-like elements of length $n+1$. Precisely, in every period, half the sequences have length one, one-fourth have length two, one-eighth have length three, etc., as long as the number of sequences so indicated exceeds 1. Moreover, due to the previous observation, the total number of sequences of 1's equals the total number of sequences of 0's, because the sequences of these two types alternates.
 - Has a two-level auto-correlation function, $C(\tau)$. Explicitly,

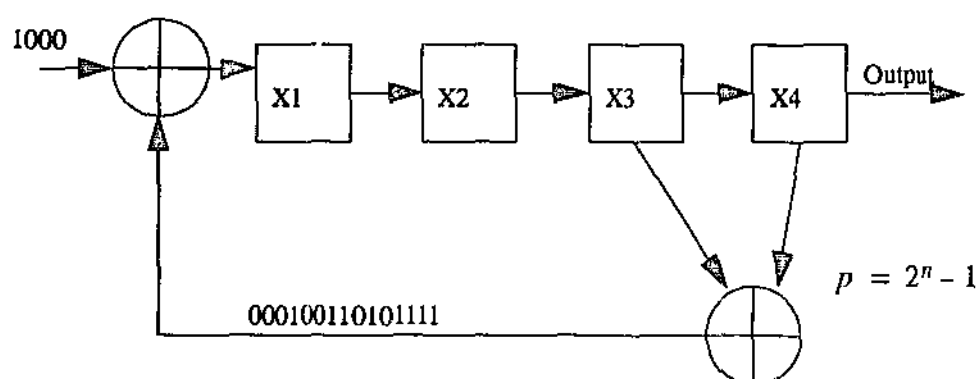
$$pC(\tau) = \sum_{n=1}^p a_{n+\tau} a_n = \begin{cases} p & ; \tau = 0 \\ K & ; 0 < \tau < p \end{cases} \quad (11)$$

Therefore, $\{a_k\}$ is a PN binary sequence *if and only if* it is a binary sequence which satisfies a linear recurrence [4]:

$$a_k = \left(\sum_{i=1}^n C_i a_{k-i} \right) \text{modulo } 2 \quad (12)$$

and has a period $p = 2^n - 1$. The n is referred to as a degree of the PN sequence $\{a_k\}$ [14].

It is shown [14] that such a PN binary sequence can be generated using shift registers with feedback, as shown in Figure 2.



n - number of shift registers

p - length of sequence

Figure 2: PN binary sequence generation using shift registers with feedback. ($n = 4$)

1.4 Constraints and limitations of DS-CDMA.

Because the implementation of CDMA uses low correlation codes to separate different signals transmitted in the same channel, the limitations to the multiple-access are:

- The number of low cross-correlation code sequence used. The maximum number of simultaneous users of the same wideband channel in CDMA is limited by the maximum available low correlation code sequence used in the system.
- The jamming margin of the channel. Jamming margin, JM , in its strictest terms, is the ability of a receiver to provide a useable receiver output when an undesired signal is present [11]. Jamming margin in a direct sequence system is expressed as

$$JM = PG - \left(\frac{S}{N_{out}} + \text{internal noise} \right) \quad (13)$$

where PG - Process Gain (refer formula 9)

internal noise - internal processing losses at the receiver

$\frac{S}{N_{out}}$ - signal-to-noise ratio after processing by the receiver

where,

$$\frac{S}{N_{out}} = \frac{S}{N_{in}} \times PG \quad (14)$$

where $\frac{S}{N_{in}}$ - signal-to-noise ratio received at the receiver

and in decibels,

$$\frac{S}{N_{out}} = \frac{S}{N_{in}} + PG \quad (15)$$

Assuming an ideal receiver, with no internal noise, with formula (13), one can derive the CDMA capability in terms of the number of users possible in a particular application [11].

Example 2:

For example [11], given a system with a 1.25 MHz spread spectrum bandwidth and a data rate of 9.6 Kbps, with required output signal-to-noise ratio of 6.0 dB, and no internal losses, jamming margin would be

$$JM = 10\log\left(\frac{1.25 \times 10^6}{9.6 \times 10^3}\right) - 6.0 = 15.1\text{dB}$$

and the maximum number of users would be

$$10^{\frac{15.1}{10}} = 32.35$$

if and only if the signal power arriving at the receiver from every user is precisely the same as the signal power arriving at that receiver from every other user [11].

This can only be achieved by [11]

- Precisely controlling the transmitted power from every user, so that all signal power arriving at a given receiver is equal.
- Controlling the distance of all users so that their arriving signal power at a given receiver is equal.
- Some combination of the above.
- Last but not least, the channel capacity as illustrated in Shannon's formula (1) for channel capacity, clearly shows a channel capacity limitation for all transmission techniques, including the DS-CDMA as well. From formula (5), we can see that for a fixed signal-to-noise ratio and a bounded bandwidth channel (even with the wideband channel there is still a finite boundary), the channel capacity for error-free transmission is also fixed. Hence, any increase in number of users beyond the maximum supported by the channel capacity, C , would increase the probability of transmission errors.

1.5 Advantages of using DS-CDMA.

DS-CDMA is selected for the following reasons:

- Since the useful signal is spread out over a broad spectrum, selective jamming or fading of the spread spectrum channel would only result in small loss in the recovered signal spectra power (Figure 3). With digital communication, as long as this recovered power spectrum is above a required threshold, no data is lost. Furthermore, any intentional jamming of the signal would have to spread its jamming signal power across the possible wideband frequency, thus greatly reducing the effectiveness of such a jamming. In the application of an indoor microwave LAN, channel fading or unintentional jamming usually occurs selectively, hence spread spectrum is a definite choice for transmission channel modulation under such conditions.
- Selective addressing capability. It is possible to recover a specific narrowband spectrum from a spread spectrum with noise. The 'noise' could be other spectrum spread signals. Hence, as long as different signals are spread with different low correlation functions, each signal could be specifically recovered from the spread spectrum by despreading each signal with its known CDMA low correlation function.
- Support for multiple access via the use of low correlational spreading sequences. Even though more than one access is spread over the same spectrum at any particular instance, the low cross correlational properties of the low correlation CDMA functions, e.g. like the Rademacher-Walsh series of orthogonal functions, ensure low inter channel interference.
- Low density power spectra for signal hiding. As the energy spectrum of the transmitted signal is spread over a wide frequency, it is possible for the transmitted signal to be hidden by much stronger channel noise. This deters possible "unauthorised" scanning of the

transmitted signal. Furthermore, by modulating the transmitted signal with a pseudorandom sequence, the signal is not only spread out but it is also 'scrambled' by the spreading PN sequence. Thus inherently, CDMA, like other spread spectrum technique, provides a 'secure' channel for transmission. This 'security' is usually not adequate as the entire spreading sequence (the spreading sequence together with the low correlation CDMA multiplexing code) of a system is finite in length. Because of this finite length, it can be 'cracked'. Usually data encryption is also used in channels requiring better security from unauthorised users.

- No equaliser needed. When the transmission rate is much higher than 10 kbps in FDMA and TDMA an equaliser is needed to reduce the intersymbol interference caused by time delay spread. However, in CDMA normally only a correlator is needed at the receiver end, to recover the desired signal from the spread spectrum signal. The correlator is usually simpler to implement than the equaliser [15].
- No guard time in CDMA. Guard time is required in TDMA between time slots. Guard time does occupy the time periods of certain bits. These wasted bits could be used to improve quality performance in TDMA. In CDMA guard time does not exist [15].
- Less fading. Less fading is observed in wideband signal propagation in a multipath radio environment. It is more advantageous to use a wideband signal in urban areas and for indoor applications than in suburban areas [15]. Hence, CDMA is a natural spread spectrum waveform suitable for microcell and in building because it is susceptible to noise and interferences [12].
- No hard hand-off. Since every cell uses the same CDMA wideband channel, the only difference is the unique low correlation CDMA sequence allocated to each of the mobile terminal. Thus, there is no hand-off from one frequency to another as it moves from one

cell to another. However a change of assigned low correlation CDMA code would occur as the mobile terminal moves from one cell to the next one. This is called a soft hand-off [11].

- On a LAN, each station is typically only transmitting a portion of the time. Hence, the CDMA capacity is expressed in terms of the number of active stations at any one given time, not the number of total stations. CDMA is advantageous for this type of network because it requires no synchronisation of the multiple communication sessions occurring at any given time. It also naturally takes advantage of the low duty cycle of transmission of the stations [12].

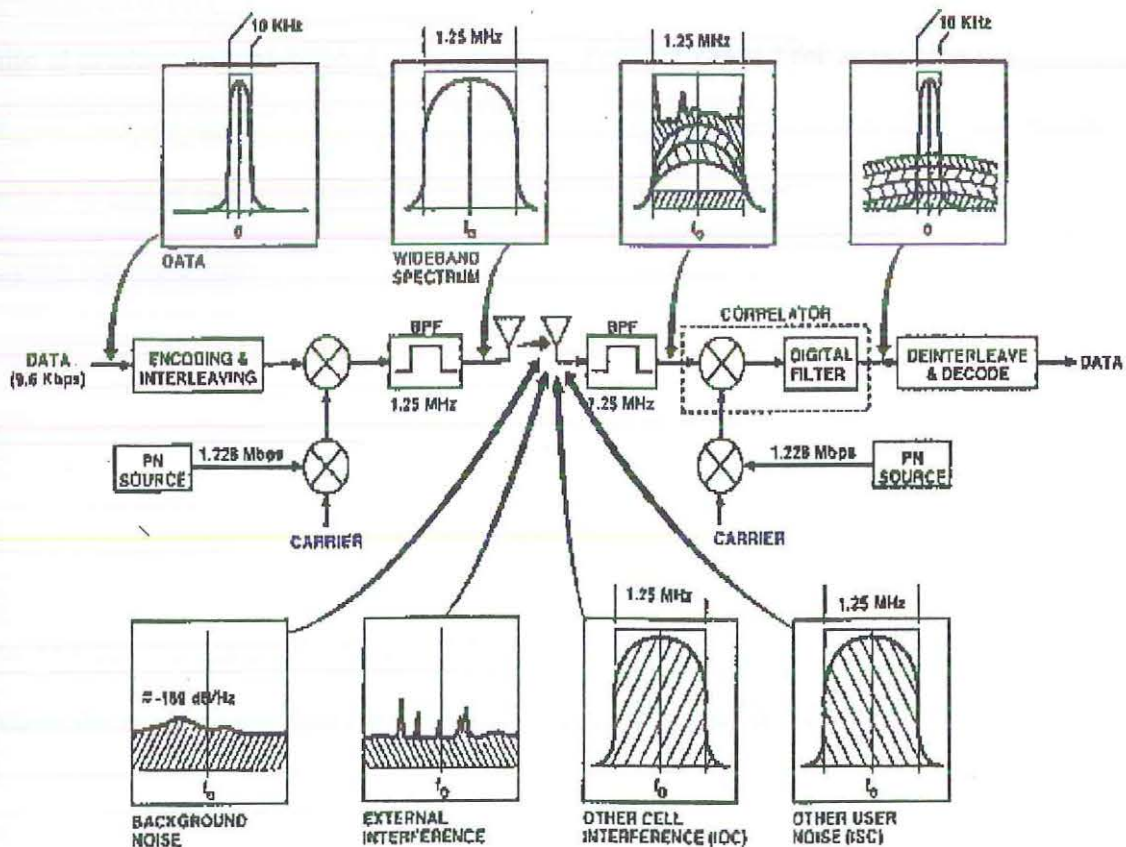


Figure 3: Illustrations of spread spectrum modulation and demodulation, taking into account the effects of channel noise [13].

Previous are only but a few advantages of using DS-CDMA for a wireless ATM LAN.

1.6 Low Correlation Functions

When a receiver uses matched filter detection - correlator, the output of the detector is the correlation function of the local reference signal and the desired signal. When several communicators are using the same channel without phase coherence, then it is desirable that the cross correlation functions to have small absolute values.

While in practice, signal defined for continuous time are needed for asynchronous communication systems, the application of a sampling theorem reduces the signal design problem to one of selecting sets of vectors $\{ (a_1^v \dots a_L^v) : v=1, \dots, M \}$ whose components a_i^v are complex numbers with

$$c_{vv} = \sum_{i=1}^L |a_i^v|^2 = 1 \quad (16)$$

Thus here lies the great importance of the cross-correlation function; it is realised that by forming the cross-correlation function among a set of M vectors of length L and norm 1,

$$c_{vr} = \lim_{L \rightarrow \infty} \frac{1}{2L} \sum_{i=-L}^L a_i^v a_i^r \quad (17)$$

a decision can be made whether the desired signal of the form a_i^v is also available in the form a_i^r , which is the function of the received signal. This is the case when $c_{vr} \neq 0$. In fact, a maximum value of c_{vr} , $\max |c_{vr}|$ results, corresponds to the phase shift of a_i^r against a_i^v . A subset of low correlational functions which exhibits ideal cross-correlation characteristics is the subset of orthogonal functions.

Another possible choice of correlation functions is the Gold codes having good autocorrelation properties to ease code acquisition.

1.6.1 Orthogonal functions

A series $\Phi_n(t)$ where $(n = 0, 1, 2, \dots)$ is said to be orthogonal over the interval $0 \leq t \leq T$ if

$$\int_0^T \Phi_n(t) \Phi_m(t) dt = \begin{cases} K & ; n=m \\ 0 & ; n \neq m \end{cases} \quad (18)$$

if $K = 1$, then $\Phi_n(t)$ is said to be an *orthonormal function*. Hence, if we transmit the product of, a signal $s(t)$, and an orthonormal function $\Phi_n(t)$, we can recover the signal component $s(t)$, at the receiver side by multiplying the received signal with the same orthonormal function, $\Phi_n(t)$, followed by an integration of this product over the fundamental period of $\Phi_n(t)$. As observed in formula (18), any other transmissions (i.e. other transmitted signals, like the product of $s(t)$ and $\Phi_m(t)$, where $n \neq m$), when multiplied with $\Phi_n(t)$ and integrated from 0 to T would give a result equal to zero. Thus, with the use of orthogonal functions, multiple transmitted signals in a wideband channel can be selectively received.

1.6.2 Rademacher-Walsh functions

Rademacher functions constitute a set of incomplete but orthogonal functions [17].

Rademacher functions can be represented by a series of square-waves having unit mark-space ratio.

$$R(n, t) = \text{sgn} \{ \sin(2^n \pi t) \} \quad (19)$$

The first function, $R(0, t)$, is equal to 1 for the entire finite interval $0 \leq t \leq T$. The next and subsequent functions are square-waves having odd symmetry. The incompleteness of the series can be demonstrated if we consider the summation of a number of Rademacher functions. This composite waveform will also have odd symmetry about the centre. However, similar, even symmetry functions required for completeness cannot be developed [17].

Rademacher functions have two arguments n and t such that $R(n, t)$ has 2^{n-1} periods of square-wave over a normalised time base $0 \leq t \leq 1$. The amplitudes of the functions are +1 and -1. Figure 4 shows a set of Rademacher functions.

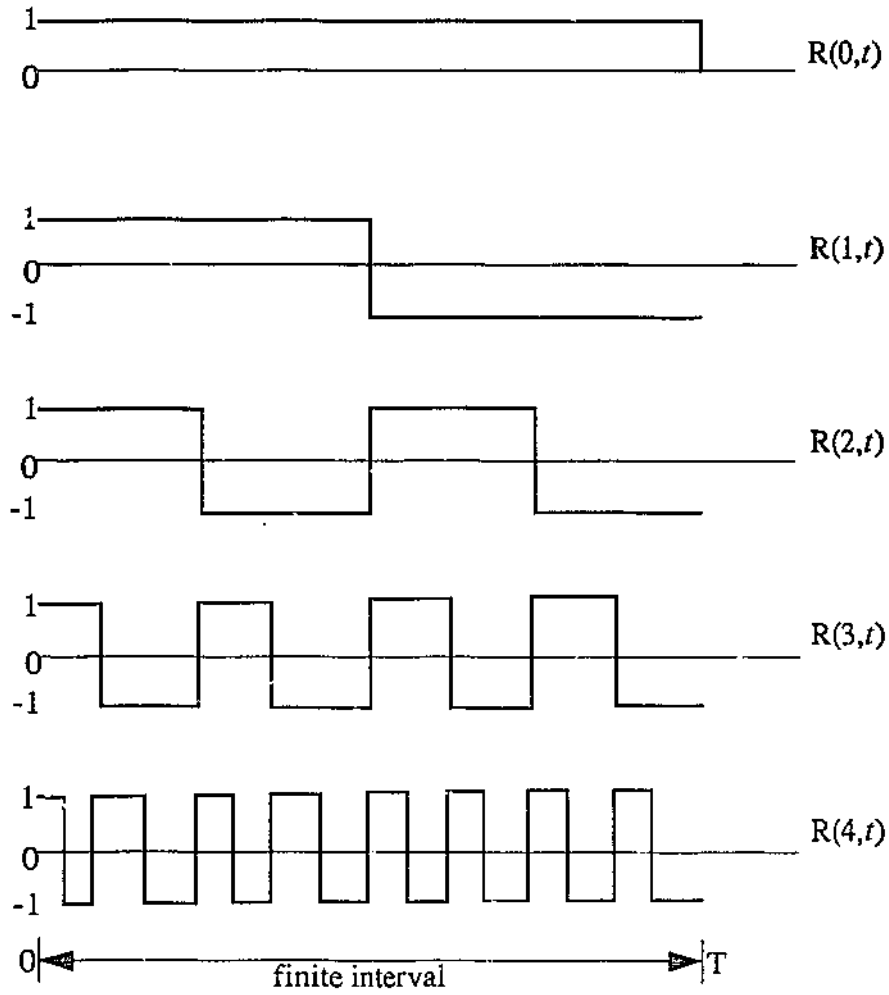


Figure 4: A set of Rademacher functions.

They can be derived from sinusoidal functions which have identical zero crossing positions and may be obtained from a sinusoidal waveform of appropriate frequency by amplification followed by hard limiting [17].

Another set of orthonormal functions are Walsh functions. The Walsh functions form an ordered set of rectangular waveform taking only amplitude values +1 and -1. Unlike

Rademacher functions, the Walsh rectangular waveform do not have unit mark-space ratio. They are defined over a limited time interval, T , known as the time-base, which requires to be known if quantitative values are to be assigned to a function. Like the sine-cosine functions, two arguments are required for complete definition, a time instant, t , (usually normalised to the time-base as $\frac{t}{T}$) and an ordering number, n , related to frequency via *sequency* - a way which is later described. The function is denoted as:

$$wal(n, t) \quad (20)$$

and for the most purposes a set of such functions is ordered in ascending value of the number of zero crossings (*sequency*) found within the time-base [17]. The motivation behind the use of orthogonal Walsh functions follows from the fact that the product of two Walsh functions yields another Walsh function [18]:

$$wal(h, \theta) wal(k, \theta) = wal(r, \theta) \quad (21)$$

Further expansion of formula (21) leads to the following

$$wal(h, \theta) wal(k, \theta) = wal(h \oplus k, \theta) \quad (22)$$

given when h and k are represented as binary numbers.

From formula (22) we can see that the product of two Walsh function can be achieved by modulo-2 adding each of the respective two Walsh function arguments (where k and h are written as binary numbers and are added according to the rules

$0 \oplus 1 = 1 \oplus 0 = 1, 0 \oplus 0 = 1 \oplus 1 = 0$ (no carry)). As Walsh functions are orthogonal, the product of two Walsh terms

$$\sum_{t=0}^{N-1} wal(m, t) wal(n, t) = \begin{cases} N & ; n = m \\ 0 & ; n \neq m \end{cases} \quad (23)$$

Hence, to extract a signal from a binary-Walsh orthogonal encoded signal, we just have to XOR that signal with its known Walsh function. All other Walsh function encoded signals which did not use that particular Walsh function would just return a zero.

A complete set of Walsh functions in natural order (dyadic order, Paley order) can be obtained from selected Rademacher function products [17]. This is the most used method of deriving Walsh functions in hardware generators because Rademacher functions can be generated using binary counters. Such a Walsh function generator was described by Harmuth [18].

Rademacher-Walsh (RW) functions have been shown to give optimal [19] short-range (i.e. indoor) CDMA codes in the sense that they offer the least sensitivity to time base error due to propagation delay differences. Unfortunately, they do not have a good power spectrum density (PSD). One way of improving the PSD is by spreading the RW functions with a long PN code sequence (this is done by DS spreading PN sequence), while suffering the drawback of increasing the sensitivity to time base error [19].

1.6.3 Generation of orthogonal Rademacher-Walsh functions.

A Walsh function generator having minimum orthogonality error is described by Besslich [20]. A detailed explanation of the logic working of such a generator is shown in [20]. It is suffice to point out here that this generator functions as a sequential network and uses Rademacher functions to derive the complete set of orthogonal Walsh functions. This method [20] has a minimised error in orthogonality as any differences in time alignment would be due to the tolerances in the switching time of flip-flops, which sets the practical limit of orthogonality.

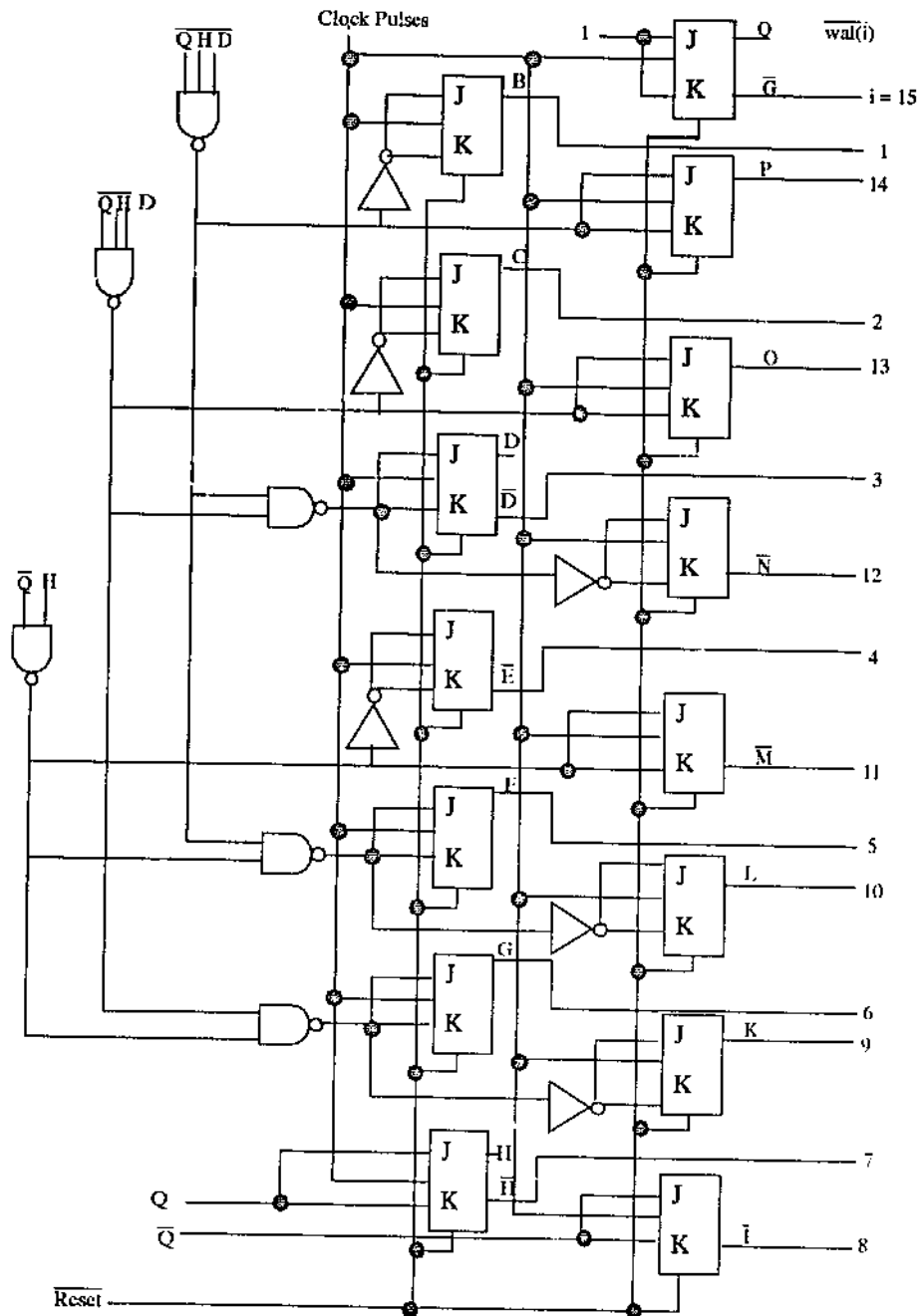


Figure 5: Diagram of generator using NAND gates and JK flip-flops [20].

The generator illustrated in Figure 5 can be implemented in COSSAP as long as it is possible to simulate the operation of a JK flip-flop in COSSAP.

A JK flip-flop can be replaced by a well known master-slave configuration of logic gates given in Figure 6.

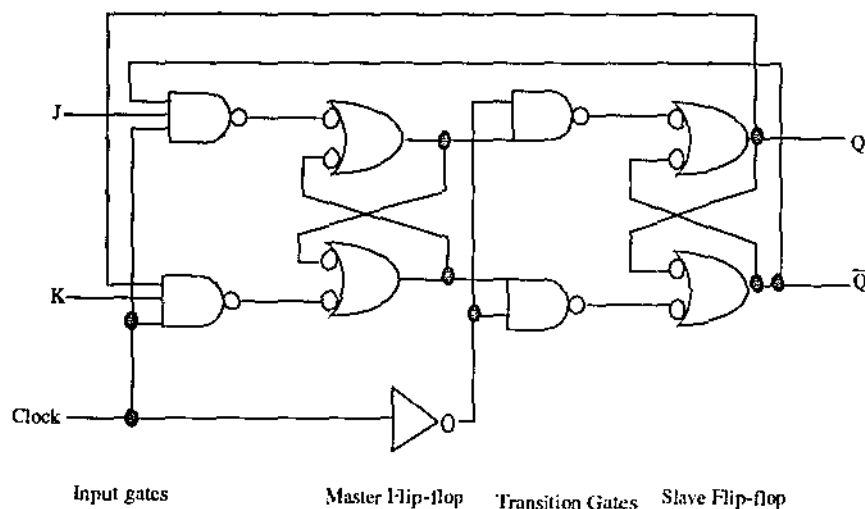


Figure 6: Master-slave equivalent circuit [20].

Such logic gates can be easily simulated with system blocks within the COSSAP environment.

Unfortunately, such a generator requires the synchronisation between transmitter and receiver. Alas, our ATM LAN operates in an asynchronous manner, hence a more suitable method of Walsh function generation has to be found.

Another approach is to 'hard-wire' each of the orthogonal Walsh functions to a specific periodic pulse generator. This way, the generation of Walsh functions in both the transmitter and receiver are independent from one another, achieving the asynchronous result as desired. Furthermore, if changes need to be made (i.e. addition of orthogonal functions to increase number of user access), the changes could be easily achieved by just changing the periodic wave generated by the system's periodic pulse generator.¹ By 'hard-wiring' the Walsh functions into the system, one also achieve another system requirement - the ability to change

the available access channel in the simulation model. An illustration of orthogonal Walsh functions achievable with a 16 bits periodic pulse is given in Figure 7 and 8.

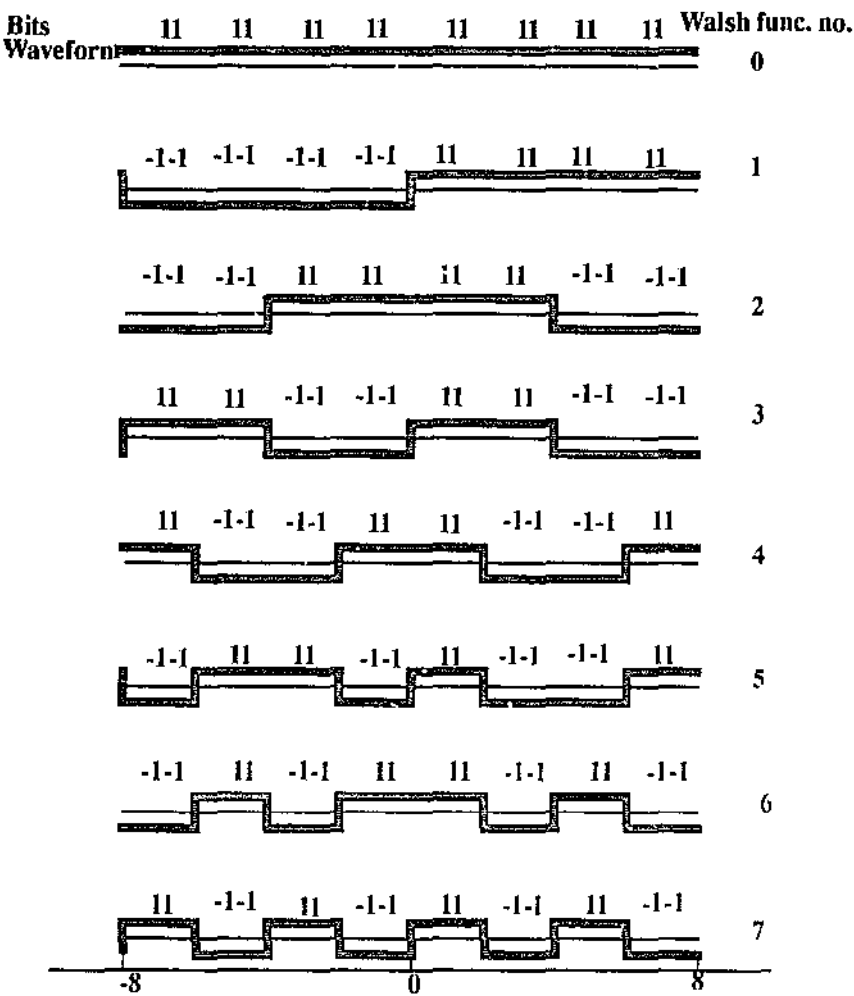


Figure 7: Orthogonal Walsh functions representation in waveform and equivalent bits.

1. The length of the periodic pulse determines the number of available orthogonal Rademacher-Walsh functions, i.e. a 16 bits periodic pulse provides 16 orthogonal Rademacher-Walsh functions. Hence, to increase the number of orthogonal Rademacher-Walsh functions in a system, we need to increase the length of the periodic pulse generated, thus increasing the number of periodic pulse generator used.

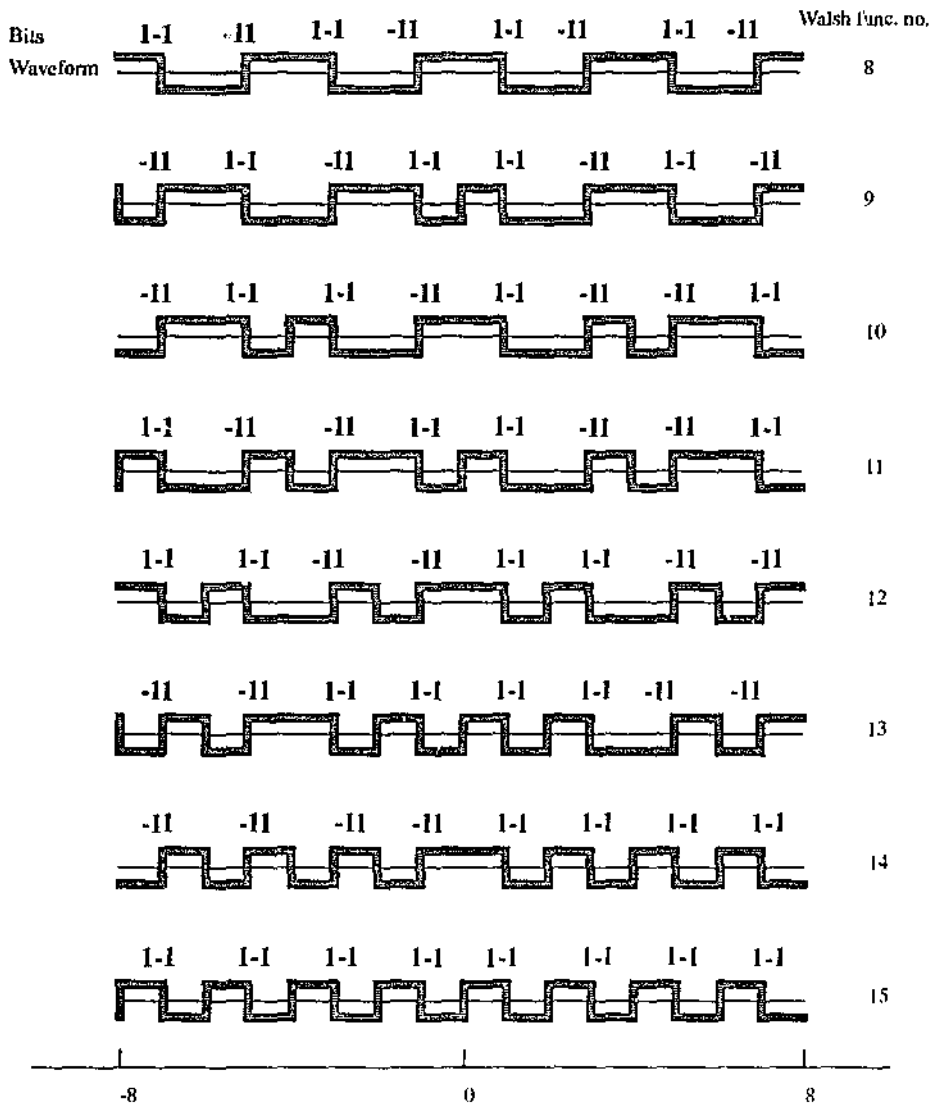


Figure 8: Orthogonal Walsh function representations. (Continue from Figure 7).

1.7 Indoor Microwave Channel Characteristics

As the transmission channel uses the microwave radio spectrum, a brief description of the basic characteristics of such a channel is helpful in designing a possible system topology.

Microwave signals are easily blocked by metal sheets, hence for the indoor local area network, the coverage of one base station cell on one floor would not interfere with the coverage sector on the next floor due to the high microwave attenuation by the steel structure of modern buildings [21].

Two dominant effects of microwave propagation are attenuation and delay [12]. The attenuation effect is due to, among other factors, the spreading of radiation. Attenuation due to this spreading varies so slowly with frequency that it can be safely considered virtually fixed within the signal bandwidth. Other possible attenuation could be due to environment factors, i.e. rain attenuation, mechanical factors, i.e. antenna mispointing and multipath fading.

As the purpose of this thesis is to simulate an indoor wireless microwave ATM LAN in COSSAP, the detailed explanation of the channel characteristics of microwave transmission is beyond the scope of this work.

From this thesis's point of view the most important fact of the indoor microwave channel properties is the frequency selective nature of the fading. Therefore, wideband signal fading is not as severe as narrowband signal fading because wideband signal takes advantage of the natural frequency diversity over its signals [15].

Multipath fading is the effect of two or more ray paths of the electromagnetic waves travelling from the transmitter to the receiver. This can be caused by inhomogeneous in the atmosphere or reflection of radio waves off obstacles, such as walls and furniture [15]. Unlike rain

attenuation (unlikely for indoor environment), which also varies with frequency, multipath fading can result in a large frequency-dependent attenuation within the narrow signal bandwidth. This phenomenon is known as *selective fading*. The mechanism for multipath fading is shown in Figure 9.

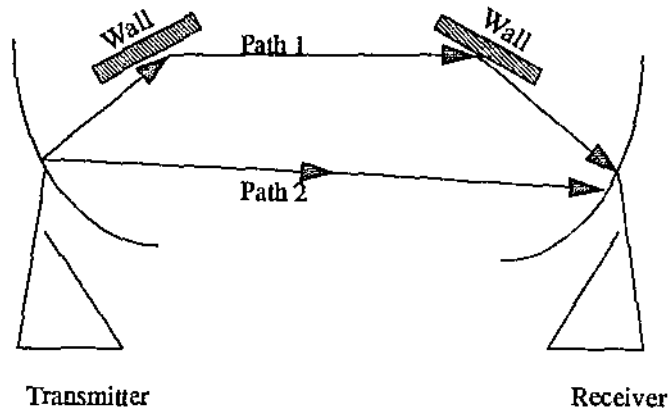


Figure 9: Illustration of two ray paths between a transmitter and a receiver resulting in different propagation delays [15].

The worst scenario would be when Path 1 totally cancels out Path 2 as the receiver will perceive only the sum of the signals.

Indoor multipath fading for transmission path with line-of-sight has the characteristics of a Rician fading channel. Rician fading occurs if the central limit theorem can be applied for the accumulations of in-phase components and quadrature components of each reflections. This occurs if the number of reflections is large and none of the reflections substantially dominates the joint reflected power [22].

Multipath fading can be mitigated using diversity technique, in which two or more

independent channels are somehow combined [15].

1.8 ATM Systems

In ATM systems, all information to be transferred is packed into fixed-size slots called *cells*. These cells have a 48 octet information field and a 5 octet header. Whereas the information field is available for the user, the header field carries information that pertains to the ATM functionality itself, mainly the identification of cells by means of a label.

ATM allows the definition and recognition of individual communications by virtue of the label field inside each ATM cell header; in this respect, ATM resembles conventional packet transfer modes. Like packet switching techniques, ATM can provide a communication with a bit rate that is individually tailored to the actual need, including time-variant bit rate.

For our purpose, the ATM system uses a minimum bit rate of 2 Mbps.

1.9 Channel Modulation Techniques

Due to the multipath fading and signal attenuation experienced by a wireless microwave channel, most of the amplitude modulation schemes could not be used. Furthermore, the multiple access nature of a spread spectrum system also made the use of frequency modulation undesirable. Hence, even though a multipath fading channel inherently displays a certain degree of phase distortion in proportion to the degree of deep channel fading, phase modulation schemes still present an attractive collection of possible channel modulation

techniques.

A family of phase modulation constellations with a simple geometry is based on phase-shift keying, sometimes combined with amplitude modulation. These channel modulation techniques - belonging to the family of M-ary Phase Shift Keying (MPSK) - are found to be effective when used in wireless multipath fading channels [12]. IEEE's 802.11 Committee even recommended the use of Binary Phase Shift Keying (BPSK) for wideband channels with a data rate up to 1 Mbps. Any data rate beyond that and up to 2 Mbps, channel modulation using Quadrature Phase Shift Keying (QPSK) is recommended [1].

1.9.1 Phase Shift Keying

In BPSK the phase of a carrier is switched between two values according to two possible messages m_1 and m_2 . The two phases are usually separated by π radians. BPSK corresponds to the two possible transmitter waveforms

$$S_{BPSK}(t) = \begin{cases} S_1(t) = -A \cos(\omega_0 t + \theta_0), & 0 \leq t \leq T_b, & m_1 \text{ is sent} \\ S_2(t) = +A \cos(\omega_0 t + \theta_0), & 0 \leq t \leq T_b, & m_2 \text{ is sent} \end{cases} \quad (24)$$

The transmitted signal is therefore equivalent to a double sideband suppressed-carrier amplitude-modulated waveform where the information signal is a digital waveform with polar format [12].

Consider two BPSK systems, one using a carrier $\sin (\omega_0 t + \theta_0)$ and the other using $\cos (\omega_0 t + \theta_0)$. By combining these two into a single equivalent system, we form a means of doubling the rate at which the data bits are transmitted over the same channel on the same carrier. The resulting system is called Quadrature Phase Shift Keying (QPSK). The equivalent transmitted signal is the sum of the two separate waveforms

$$S_{QPSK}(t) = \pm A \cos (\omega_0 t + \theta_0) + (\pm A) \sin (\omega_0 t + \theta_0), \quad 0 \leq t \leq T_b \quad (25)$$

The factor $\pm A$ of the $\cos (\omega_0 t + \theta_0)$ term represents the data from one source, whereas the $\pm A$ on $\sin (\omega_0 t + \theta_0)$ represents data from the second source. In each case the upper sign goes with a binary 0, and the lower sign corresponds to a 1. It is readily shown that formula (25) can be written as

$$S_{QPSK}(t) = \sqrt{2}A \cos (\omega_0 t + \theta_0 + \theta_k) \quad (26)$$

where θ_k can have values $\pm \frac{\pi}{4}$, and $\pm \frac{3\pi}{4}$ according to the messages. The possible amplitudes and phases (θ_k) of $S_{QPSK}(t)$ are shown as points in Figure 10.

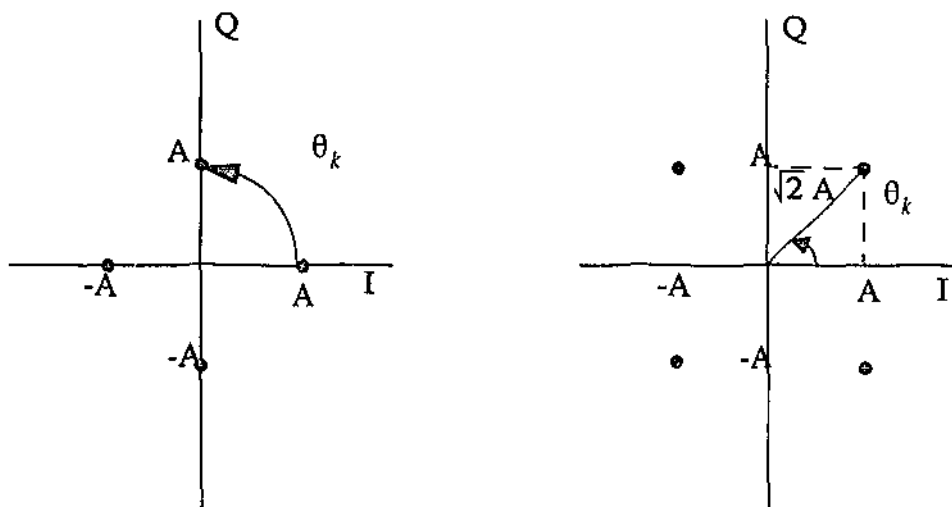


Figure 10: QPSK signal constellations (permitted values of the complex envelope).

Diagrams like in Figure 10 are called *signal constellations*.

1.9.2 Differential Phase Shift Keying

Most of the signal constellations used in modulation are seriously disadvantaged by the fact that they are rotationally invariant for some angles of rotation, typically multiples of 0.5π . If the constellation is rotated, the receiver would not be able to distinguish it from a valid constellation, unless the receiver knows the actual transmitted data symbols, which it does not. This problem can be solved by using *differential encoding*, in which the information is encoded by the *change* in the constellation position between symbols rather than absolute position [12].

Differential encoding is especially valuable in channels with rapid fading, such as mobile microwave channel. It is common practice to use Phase Shift Keying (PSK) modulation for such channels, which makes the detection of data symbols based on the angle and not the fluctuating amplitude of the received signal. During deep amplitude fades, phase of such a channel can also vary rapidly. To counter this effect, the phase variation can be mitigated by using PSK with differential encoding and differential detection. This would, however, lower the noise immunity of the channel. This combination of PSK, differential encoding and differential detection is called *differential phase shift keying*.

The generation of binary DPSK signals is illustrated in Figure 11.

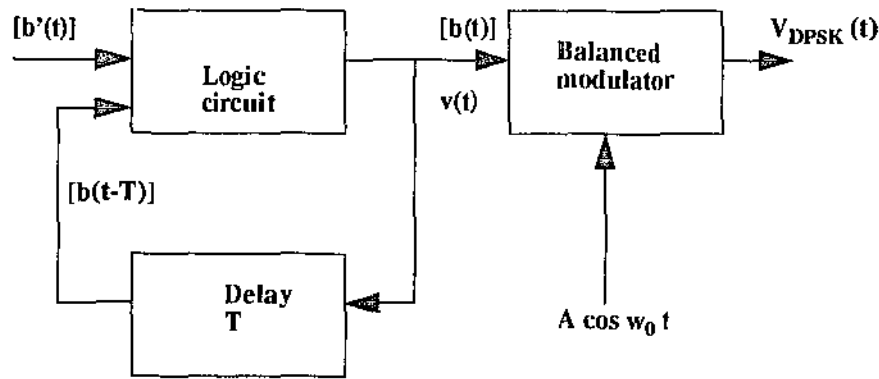



Figure 11: DPSK signal generation [14].

$$\text{where } v_{DPSK}(t) = \frac{v(t)}{V} A \cos(\omega_0 t) = \pm A \cos(\omega_0 t)$$

Sequence flow


| $b'(t)$ | $b(t)$ | Phase |
|---------|--------|-------|
| | 1 | 0 |
| 1 | 1 | 0 |
| 0 | 0 | π |
| 1 | 0 | π |
| 1 | 0 | π |
| 0 | 1 | 0 |
| 1 | 1 | 0 |
| 0 | 0 | π |
| 0 | 1 | 0 |
| 1 | 1 | 0 |

Table 1: The bit streams $b'(t)$ represents the message to be transmitted^a [23].

a. $b(t)$ is an auxiliary bit stream generated for DPSK transmission. The relative phases of the carrier in each bit interval are also indicated

The binary encoded message to be transmitted is represented by the binary sequence $b'(t)$ as shown in Table 1. An auxiliary binary sequence, $b(t)$, is generated. This auxiliary sequence has one more digit than $b'(t)$. This leading digit is arbitrary, and is taken as 1 in Table 1.

Succeeding digits in $b(t)$ are determined on the basis of the rule that when $b'(t)$ is a 1 in some bit interval, $b(t)$ in that interval does not change from its value in the preceding interval. Else when $b'(t)$ is 0, $b(t)$ does change. For instance, since the first digit in $b'(t)$ is 1, there is no change in the first two digits of $b(t)$. On the same token, as the second digit in $b'(t)$ is 0. There is a change between the second and third digits in $b(t)$ [23].

Figure 10 shows how such an encoded binary sequence could be generated. The logic circuit has two inputs, the bit stream $b'(t)$ and the bit stream $b(t)$ delayed by time T allocated to a single bit, that is, $b(t-T)$. The logic circuit performs the logical operations illustrated in Table

2.

| $b'(t)$ | $b(t-T)$ | Output $b(t)$ |
|---------|----------|---------------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Table 2: Logical operation of logic circuit.

From Table 2, we can see that the logic circuit does indeed outputs $b(t)$ from $b'(t)$ and $b(t-T)$. Furthermore, we can see that the logic circuit is no more than just performing as an inverse XOR gate.

Corresponding to the bit stream $b(t)$, a waveform $v(t)$ is generated where, say, $v(t) = +V$ for $b(t) = 1$ and $v(t) = -V$ for $b(t) = 0$. This waveform is applied as a modulating signal to a balanced modulator to which is applied, as well, a carrier $A \cos(\omega_0 t)$. The modulator output is then a carrier which changes phase when $b(t)$ changes. A change in $b(t)$ would result in a 180° phase change in the carrier output [23].

To recover the $b'(t)$ bit stream from the DPSK signal, a detector like the one shown in Figure 12 is used.

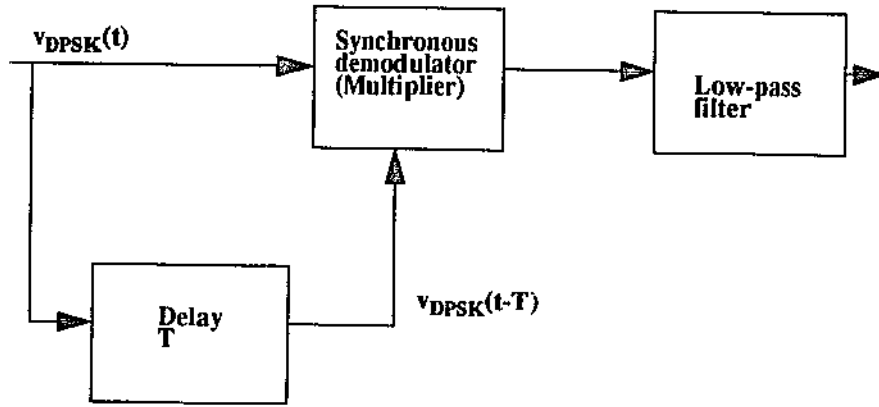


Figure 12: DPSK detector.

where $T = \frac{n\pi}{\omega_0}$

and

$$v_{DPSK}(t) = \frac{v(t)}{V} A \cos(\omega_0 t)$$

hence

$$\frac{v(t)}{V} A \cos(\omega_0 t) \left(\frac{v(t-T)}{V} A \cos(\omega_0 (t-T)) \right) = v(t) v(t-T) \frac{A^2}{2V^2} \left[\cos \omega_0 T + \cos 2\omega_0 t - \frac{T}{2} \right]$$

The product of $v_{DPSK}(t)v_{DPSK}(t-T)$ is exactly $b'(t)$ and a low-pass filter may separate this signal waveform from the double-frequency carrier term in the right-hand side of the above equation. This separation should be adequate to allow a determination, in each bit interval, of whether a 1 or a 0 was transmitted. In order to obtain a large signal output, T should be selected so that $\cos \omega_0 T = \pm 1$. Thus, the carrier frequency $0.5 \omega_0 \pi$ should be selected so that the bit duration is an integral number of half cycles in duration [23].

In comparison, DPSK has an advantage over PSK as it requires a less complicated circuitry. But it has a higher error rate, as in DPSK a bit determination is made on the basis of signal received in two successive bit intervals, noise in one bit interval may cause error to two bit

determinants. As a matter of fact, DPSK has a tendency for error to happen in pairs [24].

1.10 Proposed System Topology and Overview

The preliminary system specifications are as follow:

- The model must be able to support the transmission of ATM traffic (initially at a static transmission rate of 2Mbps but would be able to support a dynamic transmission rate if required later on).
- The model must be able to accommodate maximum number of mobile terminals with a minimum probability of error. Initially the simulation would be just on raw ATM packets, but later on, an Error Correction Coding would be included. Error detection and correction capability must be sufficient enough to support the asynchronous transmission of ATM packets.
- The maximum tolerated probability of error must be of magnitude of 10^{-7} or less.
- The wideband channel is set at 80 MHz.
- Each base station/cell has a coverage area of 20 metres in radius.
- The effective signal-to-noise ratio received at the receiving antenna must be at least 20 dB and a maximum tolerable signal power loss due to transmission path attenuation is not more than 5%.

As the area covered by the base station is $\pi \times 20^2 = 1256.64$ metres square, only one base station is necessary to cover an entire floor of a typical office building. Hence, a system topology would be as in Figure 13.

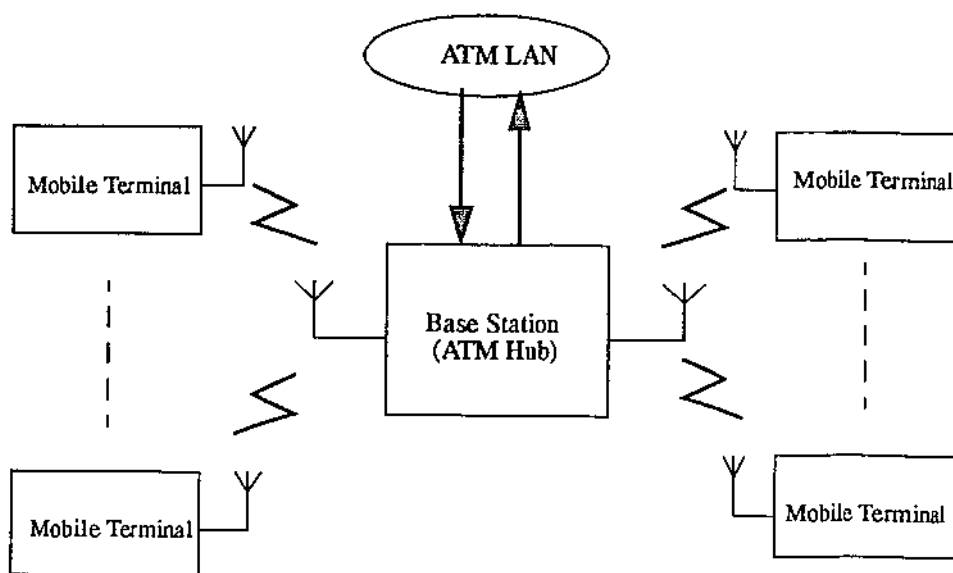


Figure 13: Overall system topology.

There would be two separate channels, one for the transmission from the mobile terminals to the base station (up-link) and the other would be for the transmission from the base station to the mobile terminals (down-link).

Not all of the available low correlation functions (in this case orthogonal Rademacher-Walsh functions) would be allocated for regular transmission access. A fix amount would be reserved for use in control channels, i.e. $wal(0, \theta)$ could be used for the pilot channel, as in Figure 14.

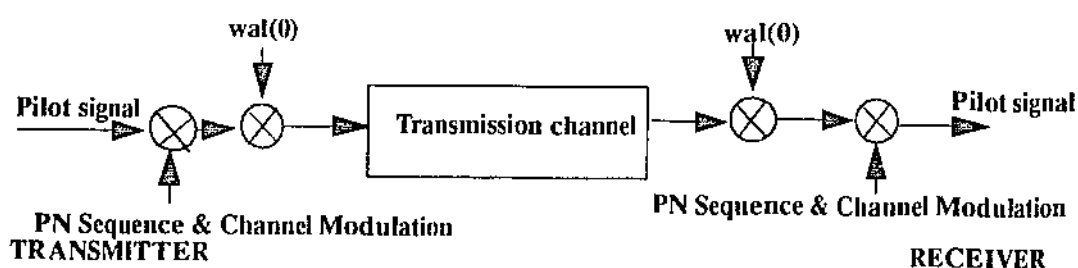
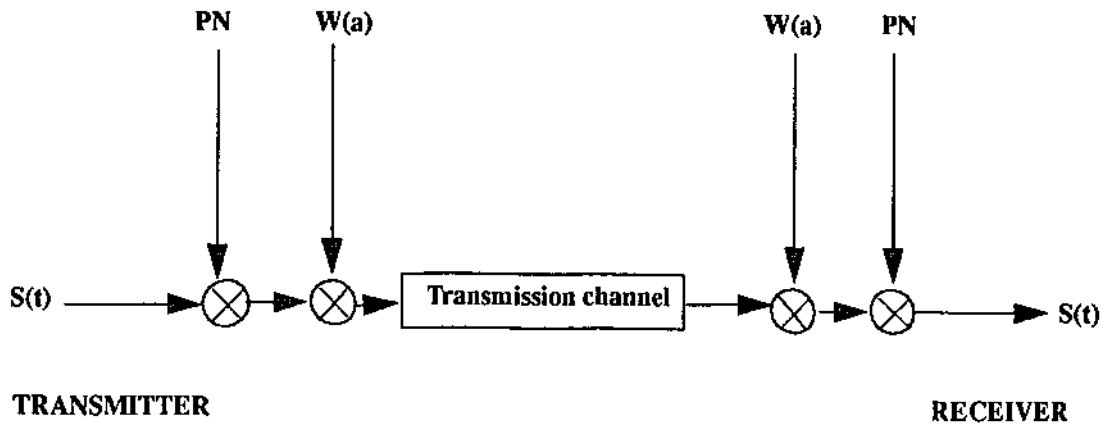


Figure 14: Pilot channel

For the regular access channel, a possible configuration would be as in Figure 15.



$W(a)$ = assigned Walsh function

$S(t)$ = Transmitted signal

PN = Pseudorandom Noise (together with Channel Modulation)

Figure 15: Access channels.

Initially, we would like to simulate a system with 16 low correlation DS-CDMA access channels, using a bipolar binary version of the 16-bits Rademacher-Walsh functions.

2 Overview of Communication System Simulation and Analysis Package

2.1 Introduction

In this chapter, basic principles in the use of COSSAP as a design and simulation tool are presented. Programming and debugging methods used in the development of the DS-CDMA transmission network model are also given and software constraints of the system are highlighted. Finally, a brief description of the hardware platform for the system is given. The issues dealt with in this chapter are only those pertinent to the design and development of the DS-CDMA indoor microwave ATM LAN physical layer simulation model.

2.2 Communication System Simulation and Analysis Package

COSSAP is an advanced, integrated Computer Aided Engineering environment for designing, simulating, and implementing Digital Signal Processing (DSP) and communication systems. COSSAP mainly interacts with a user via Graphical User Interface. Individual COSSAP modules includes [25]:

- A block diagram editor (the CONFIGURATOR) to define parameterised hierarchical models and parameterised simulation runs. This is the first module used by the user.
- A Stream Driven Simulator (SDS) engine. This module is used after models were defined but before generating any code.
- A Hardware Description Language (HDL) Code Generator (VGC) to generate code for a design created using the CONFIGURATOR and simulated in the SDS.

-
- A DSP Code Generator (DGC) to generate code for a design created using the CONFIGURATOR and simulated in the SDS.
 - A link to Very Large Scale Integration synthesis tools (AHDL Simulation Interface).

The SDS not only simulates a system, but also includes tools to display and analyse simulation results both during and after a simulation:

- An utility to display data while a simulation is running (*xdsint* utility).
- A report generator that displays text about the simulation (REPORT utility).
- A graphical analyser that depicts the simulation results (CHART utility).
- An utility to generate C code for a model from generic C code (*coins* utility).

The VCG, DCG and the VHDL Simulation Interface tools help to convert any hierarchical block diagram into actual hardware.

2.3 Building models in COSSAP

To fully modularise a design in COSSAP's design environment, it is advisable to group functional blocks into modular blocks. Two types of models can be constructed in COSSAP - hierarchical and primitive. All user constructed models should be stored within user defined libraries (this is usually under the \$COSSAP_USER_D subdirectory in the user's work space). Step by step guidance in setting up an user's library space is given in the COSSAP documentation [26, 27].

Hierarchical models are constructed from existent functional blocks in the COSSAP libraries. Example of such construction is vividly illustrated in the user documentation - the COSSAP User's Guided Tour [25].

Primitive models building involves manually generating model description file for each of the primitive models constructed. ANSI C is used in the coding of the primitive function, together with the COSSAP system calls. Usually these system calls handles all input/output and memory management routines. FORTRAN77 can also be used in coding of these primitive models. A step by step guide in building of a primitive model is also given in the user documentation [26, 27]. This is accompanied by a simple example, illustrated both in ANSI C and FORTRAN77 implementations.

2.4 Errors and debugging COSSAP programs

Usually syntax and mismatched data type error messages are very descriptive and given either at simulation or compilation time. Another type of error which is not so descriptive is the system error.

This is usually caused by design overflows due to the system's constraints. For these obscure errors usually all the user would see is the message, "*Error in simulation xxx.*" Unlike syntax or mismatched type errors, this error does not halt the compilation and execution of the simulation program. Only void output data sets are given as a result of such errors. Hence, it is a very good practice to check all simulation for such errors before performing a long parameterised simulation. System constraints are well documented in the software manuals [25] and are summarised in the next section.

The REPORT utility of the system would also display all system parameters for a simulation run. This includes indications of errors in any functional blocks and the types of error. It is an

useful tool for pinning down an ‘unknown’ error in any simulation runs. Another debugging tool, which is available only to user-built primitive models, is the single stepping debugging mode. To have this facility, the user-built primitive model’s code have to be compiled with the debug mode switched on. To execute a program in debug mode, the command line utility *xsds* has to be used while starting the simulation [25]. The simulation result can be traced using the *xsdsint* utility. Further explanation on this is given in the user documentation [25].

2.5 System constraints of COSSAP

The following tables list the system constraints of the COSSAP.

| Stream Driven Simulator | |
|---------------------------------|--|
| total number of input datasets | (no. of defined datasets)*(no. of it. + 1) ≤ 150 |
| total number of output datasets | (no. of defined datasets)*(no. of it. + 1) ≤ 1000 ^a |
| max size of scratch pad (IH,RH) | 1000*512 REALs/INTEGERs (4 Bytes) |
| max size of scratch pad (DH,CH) | 1000*512 DOUBLE/COMPLEX (8 Bytes) |

Table 3: Constraints of COSSAP SDS

- a. The total number of output datasets is dependent on the number of iterations. For example if you run only one iteration, the maximum number of output datasets that you can use must be less than or equal to 500:

$$(500) * (1 + 1) \leq 1000$$

| COSSAP model libraries | |
|---|----|
| total number of system and user libraries | 64 |

Table 4: Constraints of COSSAP model libraries

| COSSAP model libraries | |
|--|------|
| max number of model classes | 128 |
| total number of models in all libraries (no limit for one library) | 3072 |

Table 4: Constraints of COSSAP model libraries

| COSSAP <i>.mdef</i> file | |
|---|-----|
| max number of parameters in one model | 128 |
| max number of input ports | 400 |
| max number of output ports | 400 |
| max number of lines in <i>mdef</i> file | 800 |

Table 5: Constraints of COSSAP *.mdef* file

| CONFIGURATOR Tool | |
|---|-----|
| max number of exit masks | 10 |
| max number of models in one exit mask | 50 |
| max number of pictures | 5 |
| max number of models in one config level | 400 |
| max number of signals in one config level | 400 |
| max number of edges of one signal | 256 |
| max length of filenames | 132 |

Table 6: Constraints of COSSAP CONFIGURATOR Tool

| CHART Tool | |
|-----------------------------------|----|
| max number of graphs in one chart | 12 |

Table 7: Constraints of COSSAP CHART Tool

| | |
|--|------|
| CHART Tool | |
| max number of displayed graph elements | 8192 |

Table 7: Constraints of COSSAP CHART Tool

| | |
|---|------|
| REPORT Tool | |
| max number of models evaluated for report | 4000 |

Table 8: Constraints of COSSAP REPORT Tool

2.6 Hardware Platform

The hardware platform used in the design, development and simulation of the DS-CDMA indoor microwave ATM LAN COSSAP model is a Sun SPARC LX workstation with 64 Mbytes of RAM, running Solaris 3.3. The COSSAP system is installed to run in a stand-alone mode.

3 COSSAP Implementation of a DS-CDMA transmission network

3.1 Introduction

In this chapter, the design and development of a DS-CDMA indoor microwave ATM LAN simulation model implemented in COSSAP software environment is presented. Each functional component in the simulation model is implemented as a modular functional block in the COSSAP environment. Some system components may have other interchangeable functional blocks for different system modelling parameters and these additional blocks are also presented. In addition, the simulation method used to evaluate the system BER performance of the DS-CDMA indoor microwave ATM LAN model is given.

3.2 System Model

The initial system specifications for the required ATM WLAN being modelled are listed in Table 9.

| | |
|--------------------|-------------------|
| Data Rate (Mbps) | 2 |
| Mobility | Stationary/Mobile |
| Range (m) | 20 |
| Frequency | 2.4 - 2.48 GHz |
| Wideband Bandwidth | 80 MHz |

Table 9: Initial system specifications for ATM WLAN

| | |
|-----------------------------|---------------------|
| Bit Error Rate | 10^{-7} or better |
| Signal to Noise Ratio (dBm) | -20 |

Table 9: Initial system specifications for ATM WLAN

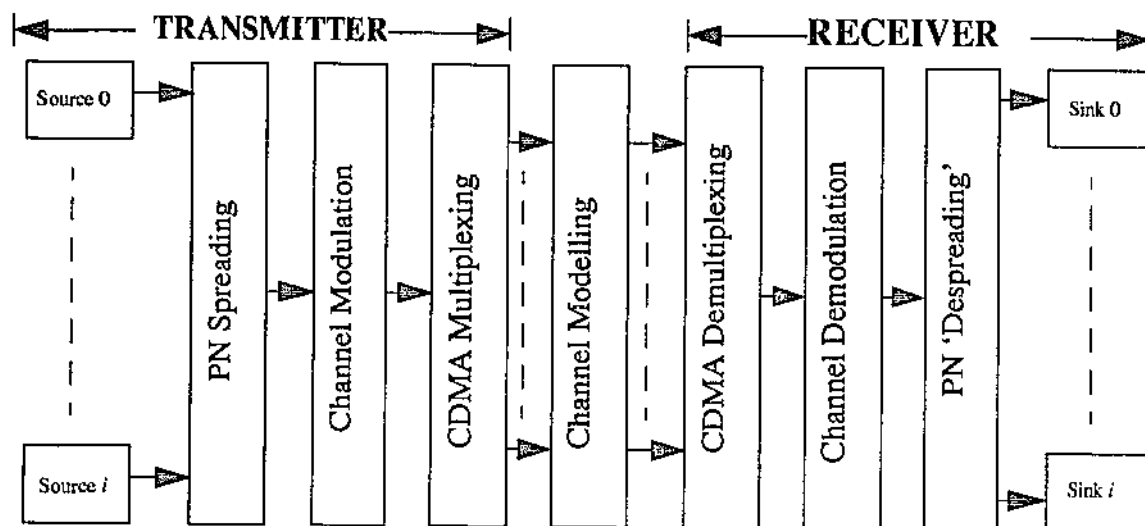


Figure 16: A simplified block diagram of the simulated system

The passband model of the system in Figure 16 was simulated and analysed in COSSAP on a Sun SPARC LX. The simulated system can accommodate up to a maximum of 16 user access channels. Actual number of simultaneous user channels which can be simulated at one time is limited only by the system constraints of COSSAP and the CDMA code set used. For the initial system, a 16-bits Rademacher-Walsh orthogonal code set is used, hence the maximum number of user access channels is limited to 16.

3.2.1 Transmitter Modelling

As indicated in Figure 16, unlike popular practice, PN spreading and CDMA multiplexing of an user access channel is separated and the latter is performed only after channel modulation to ensure proper user access channel isolation within the single wideband transmission channel. The separation of PN spreading and CDMA multiplexing, which also further spread the transmitted data, allows for the control of the spreading ratio. This is done by varying the spreading ratio of the PN sequence of the user data signal, thus allowing the user to use CDMA code set of different lengths. Nevertheless, in the initial model, the CDMA code must be equal to or less than 40 bits in length for the proposed ATM WLAN to ensure a wideband transmission channel of 80 MHz or less. Thus, the transmitted signal, $s(t)$ is expressed as [28]:

$$s(t) = \sqrt{P_s} \sum_{i=-\infty}^{\infty} (C_i \cdot D_i + jC_i \cdot D_i) g_T(t - iT) \quad (27)$$

where P_s - Average transmitted power

C_i - In-phase channel multiplexing codes

D_i - In-phase spread data symbols

$g_T(T)$ - Square-root Nyquist pulse-shape with roll-off $\alpha = 0.4$

A block diagram representation of a transmitter channel is given in Figure 17.

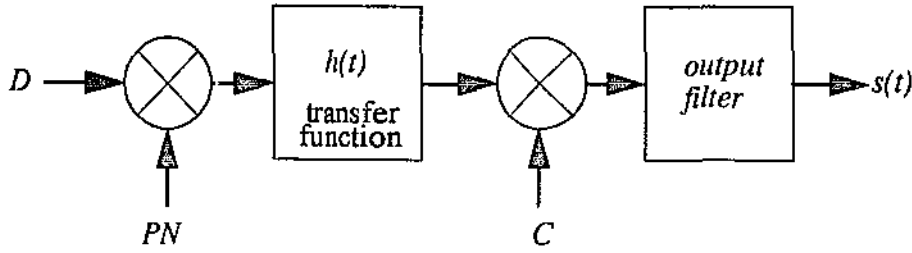


Figure 17: A block diagram transmitter channel representation

Overall Transmitter Design

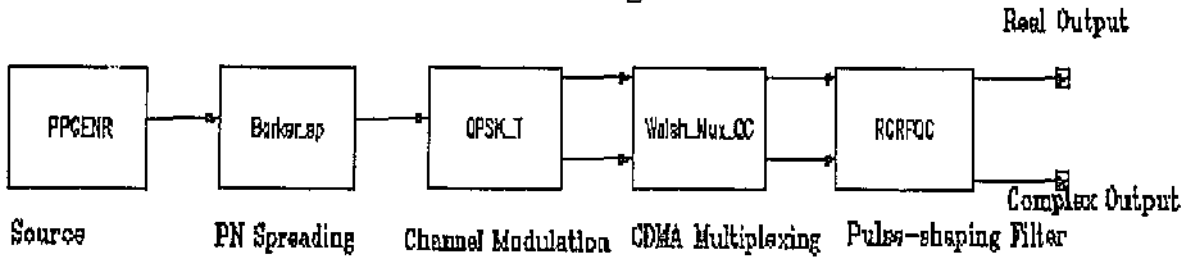


Figure 18: Initial transmitter design in COSSAP

Figure 18 shows the COSSAP functional block diagram of a transmitter. Each functional block is as follows:

1. PPGENR is a cyclic real signal generator which repetitively generates real type data bits from a user defined input dataset. For this model, bipolar real signal of a finite length is used for data source. (*Refer Appendix A*).
2. Barker_sp is a hierarchical block which spreads the user data by a spreading ratio SB_Hold . Initially, the IEEE 802.11 recommended 11-bits Barker sequence is used [1], but this can be easily changed to accommodate the use of

other possible pseudorandom spreading sequences. The $\$B_Hold$ variable determines the exact number of bits of the spreading sequence which is multiplied with each user data symbol and thus spreading it by such a ratio. Of course, this also determines the chip rate of the spreading sequence in relative to the user data rate. (*Refer Appendix B*).

3. Channel modulation is initially performed using QPSK with Trellis Coded Modulation (TCM). This initial choice is in accordance with the recommendation of IEEE 802.11 committee [1] with regard to channel modulation scheme proposed for channel with data rate between 1 Mbps to 2 Mbps. Other possible channel modulation blocks such as BPSK with and without TCM were also constructed. Evaluation of different channel modulation schemes is reduced to just merely substituting one modulation scheme block with another and simulate the new system model. (*Refer Appendix C*).
4. Walsh_Mux_QC consists of two individual CDMA multiplexer blocks - Walsh_Mux. This is because one block is used on the real part of the transmitted signal, while the other is for the complex component of the same signal. With the initial design, 16-bits Rademacher-Walsh orthogonal sequences [5] were used as the CDMA code set. Of course, other code set such as Gold codes or Kasami code set can be used by merely substituting the input datasets to the CDMA multiplexer block with these low correlation codesets. In the initial system model, the ratio of further data spreading is determined by the variable $\$Walsh_Num_Bits$ of the CDMA multiplexer block. As the initial choice of CDMA code set is the 16-bits Rademacher-Walsh orthogonal codes, thus

$\$Walsh_Num_Bits = 16$. (Refer Appendix D).

5. RCRFQC shape the data symbol pulses by a 0.5 Nyquist root raised cosine filter with a roll-off of 40%. (Refer Appendix A).

3.2.2 Channel Modelling

For an indoor microwave WLAN, the usual channels of interest are those of multipath fading, such as Rayleigh and Rician fading channels. Nonetheless, initial test are done in additive white Gaussian noise channel (*AWGN*). Within COSSAP's library of channel model blocks, all three channels are available. (Refer Appendix E). Of course for 'realistic' system modelling, some empirical channel propagation and noise measurement have to be gathered. This is to ensure that the parameters governing each channel model are set to their respective 'real' values.

By selecting a different signal power for each user in a modelled system, the effect of interferences between near-far transmitting users can also be simulated. For instance, a Rician fading channel with the possibility for signal power control (via the variable $\$Signal_Power$ in the *AWGNQC* block) is given in Figure 19. Such a fading channel modelled in COSSAP is adequate to simulate the mobile environment of a WLAN - allowing for a minimum moving velocity of 6 km/h (just about the speed of human walking).

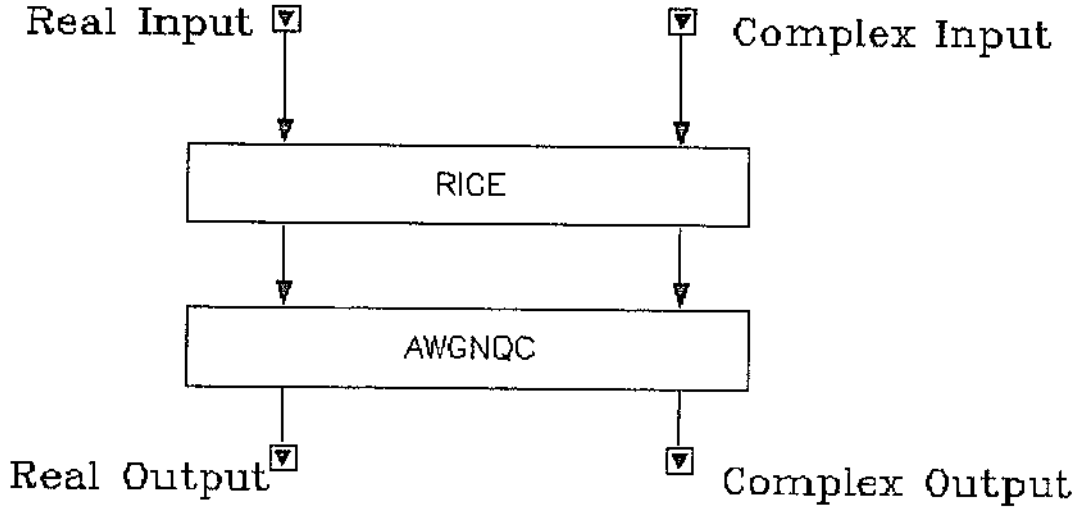


Figure 19: Rician fading channel with signal power control

The influence of noise and other user access channels within the transmission channel is modelled as linearly additive. Such modelling is possible with the use of complex signal arithmetic addition blocks available within the COSSAP libraries. In the event of more user channels than the available adders can handle individually, cascade of adders is possible in COSSAP as the accumulative law is observed by these adders. *(Refer Appendix F)*.

3.2.3 Receiver Modelling

For the initial design, a coherent detection is assumed. The received signal, $r(t)$ is therefore of the form [28]:

$$r(t) = \sum_{l=1}^N \sqrt{P_s} \sum_{i=-\infty}^{\infty} (C_l \cdot D_i + jC_l \cdot D_i) g_T(t - iT) + w(t) \quad (28)$$

where N - Number of active users

$w(t)$ - Noise, including the interference from other user channels

This leads to a block diagram channel receiver depicted in the Figure 20.

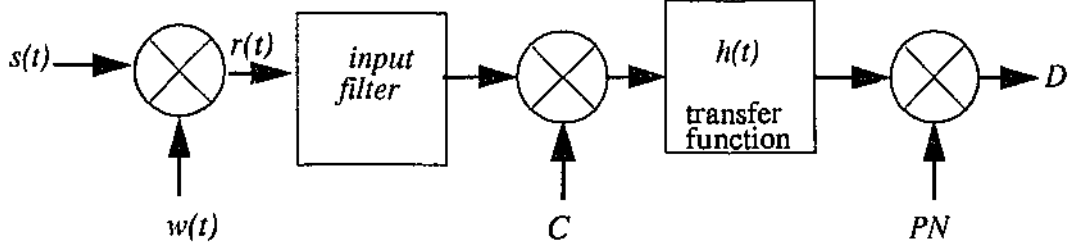


Figure 20: A block diagram representation of a receiver channel with noise

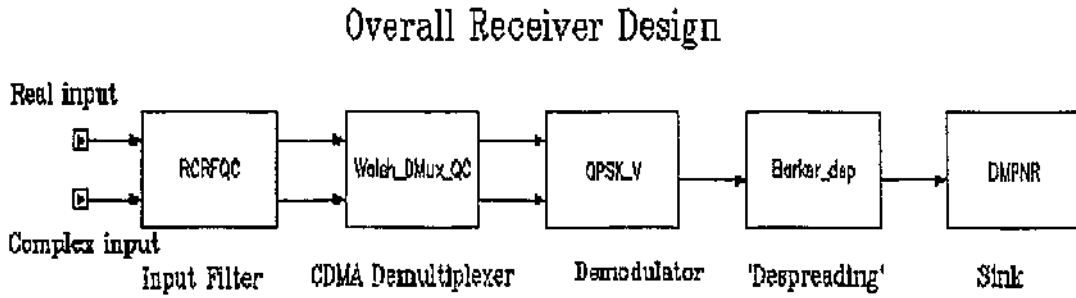


Figure 21: Initial receiver design in COSSAP

Figure 21 shows the COSSAP implementation of the channel receiver. In this initial receiver design the following COSSAP functional blocks have been used:

1. RCRFQC filter block is configured identically to the same block at the transmitter end (matched filter detection).
2. Walsh_DMux_QC filters the desired signal from all transmitted signals. This is possible due to the orthogonal nature of the CDMA codes used, where for a T length orthogonal code set $C_i(t)$ with $(i = 0, 1, 2, \dots, n)$, the following is satisfied:

$$\int_0^T C_i C_j dt = \begin{cases} K; i = j \\ 0; i \neq j \end{cases} \quad (29)$$

Hence the CDMA demultiplexer is constructed with an integrator, among other blocks. This is clearly illustrated in Figure 22.

Walsh Demultiplexer

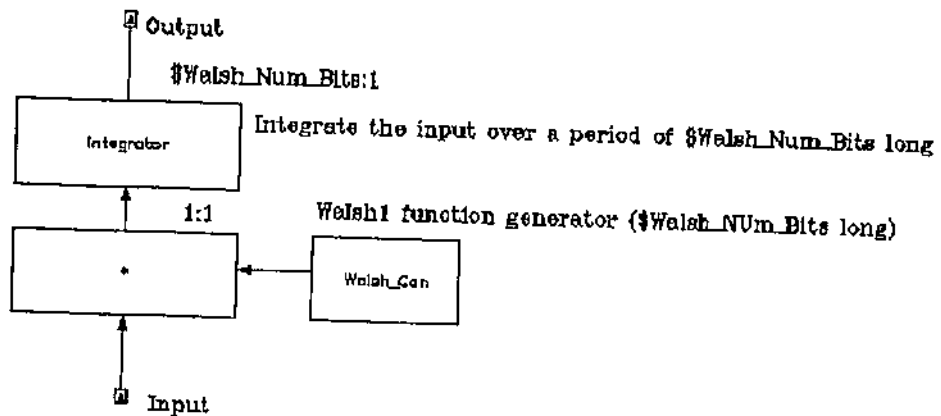


Figure 22: COSSAP block of Walsh_DMux

3. Channel demodulation block QPSK_V also included Viterbi soft-decoding. As before, channel demodulation schemes have also been developed into hierarchical COSSAP blocks.
4. Data 'despreading' is once again performed using the same 11-bits Barker sequence with decimation.
5. DMPNR operates as a received data sink and produces output datasets which can be displayed as graphs with the COSSAP CHART utility. (In some cases this sink can be replaced by a signal comparator and a BER counter). (Refer Appendix A for such a BER Counter).

3.3 Functional blocks used in the design and development of the COSSAP simulation model of DS-CDMA indoor microwave ATM LAN

All blocks have been built as hierarchical functional models. Two reasons behind the choice of using hierarchical models are:

- As much as possible, primitive blocks from the COSSAP libraries are used as these primitive models have been thoroughly tested. Furthermore, almost all extensive blocks can be constructed from the available primitive blocks.
- Construction of higher level functional blocks allows program modularity to be achieved, hence making parameter changes in resultant simulation model simple and transparent to the user.

Functional hierarchical models built so far are:

- PN spectrum spreading and despreading blocks (Appendix B)
 - Barker sequence spreading and despreading for real signals.
- Transmission channel modulator and demodulator blocks (Appendix C)
 - QPSK modulator and demodulator with the use of TCM and Viterbi soft decoding.
 - DQPSK modulator and demodulator with the use of TCM and Viterbi soft decoding.
 - BPSK modulator and demodulator with the use of TCM and Viterbi soft decoding.

-
- DBPSK modulator and demodulator with the use of TCM and Viterbi soft decoding.
 - CDMA multiplexer and demultiplexer blocks (Appendix D)
 - Walsh sequence code multiplexer and demultiplexer for real signals.
 - Walsh sequence code multiplexer and demultiplexer for complex signals.
 - Arithmetic signals adder blocks (Appendix F)
 - Arithmetic adder for 9 complex signals.
 - Arithmetic adder for 16 complex signals.

3.3.1 PN spectrum spreading and despreading blocks

In the DS-CDMA indoor microwave ATM LAN model, a 11-bits Barker sequence is used as the PN code. The 11-bits Barker sequence used is:

1, -1, 1, 1, -1, 1, 1, 1, -1, -1, -1

This is in accordance with the recommendation made by the IEEE 802.11 committee [1]. Nonetheless, other PN sequences could be implemented just by changing the input dataset of the PN spreading and despreading functional blocks. The ratio of spreading can be adjusted by

altering the holding ratio of the PN spreading block and the sampling ratio of the PN despreading block.

3.3.1.1 Barker sequence spreading and despreading for real signals

The Barker sequence spreading and despreading functional block consisted of two models, a modulator (spread) and a demodulator (despread).

In the modulator, the input signal is passed through an ideal holding (*HOLDR*) block for a user-defined spreading ratio (*\$B_Hold*). This spread signal is then multiplied with the pseudorandom 11-bits Barker sequence, to achieve a desired DS spread spectrum wideband signal.

In the demodulator, the reversed operation is performed. Firstly the input signal is again multiplied with the same pseudorandom 11-bits Barker sequence. This is to recover the spread signal component from the wideband spread signal. Next, this signal is sampled at the same ratio as it was held earlier in the Barker spreading block. The sampler used is an ideal sampling block (*SRDR*). Both the Barker sequence spreading and despreading COSSAP functional blocks are depicted in Appendix B.

3.3.2 Transmission channel modulator and demodulator blocks

For transmission channel modulation, the IEEE 802.11 committee [1] has recommended the use of BPSK for transmission of with data rate up to 1 Mbps and QPSK for transmission with

data rate up to 2 Mbps. Thus, in the considered COSSAP simulation model of the DS-CDMA indoor microwave ATM LAN, functional blocks catering for both transmission channel modulation technique were developed.

To test these recommended channel modulation schemes with their respective 'differential' counterpart, equivalent DQPSK and DBPSK COSSAP models were also built.

3.3.2.1 QPSK with Trellis Coded Modulation and Viterbi soft decoding

These QPSK channel modulation and demodulation functional blocks use the TCM in the modulator and the Viterbi soft decoding at the demodulator.

As TCM requires fixed parameters in channel encoding, this functional block is fixed to support only QPSK. The Trellis state diagram is given in Figure 23.

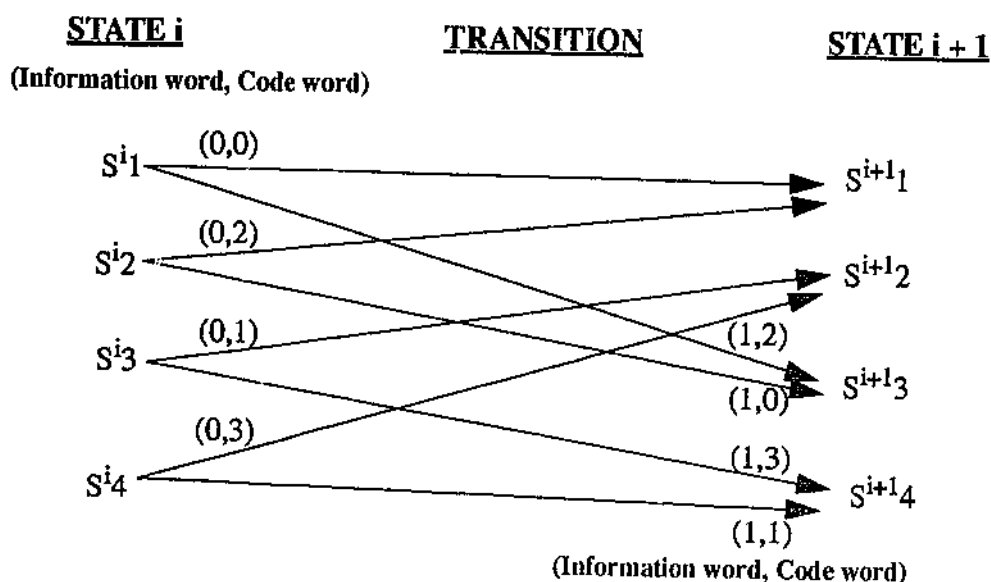


Figure 23: Trellis state diagram for QPSK

The left side of Figure 23 shows four possible encoder states with two possible state transitions from each state, that is, one information bit (two possible values). Each transition is labelled with the information word that causes this transition and the code word that is output.

The right side of Figure 23 shows the state that the encoder reaches after the corresponding transition. For instance, in state 1 with information word 1(= 1), the code word 2(= 10) is transmitted; after transition, the new state is state 3.

Because the Trellis encoder and Viterbi decoder models aim at TCM and coding schemes, code words are not transmitted directly. Instead, they are mapped onto complex channel

symbols (modulation), which are transmitted (so are also the expected input of the decoder). The QPSK constellation mapping used is given in Figure 24.

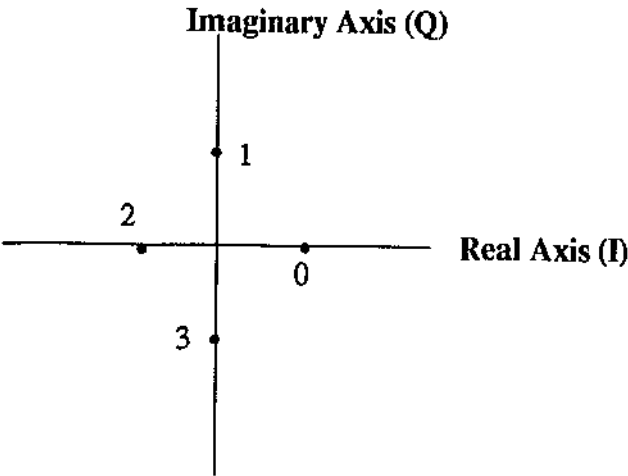


Figure 24: QPSK constellation mapping

The parameter file for the TCM and Viterbi soft decoding used in this functional block is given in Appendix C, together with the relevant COSSAP models for this channel modulation scheme.

3.3.2.2 DQPSK with Trellis Coded Modulation and Viterbi soft decoding

The COSSAP models for DQPSK are exactly similar to those for QPSK, except for the fact that in the DQPSK COSSAP models, a Differential Phase Shift Keying Modulator/Demodulator is used instead of an absolute M-ary Phase Shift Keying Modulator/Demodulator.

The parameter file for the TCM and Viterbi soft decoding used in this functional block is also given in Appendix C.

3.3.2.3 BPSK with Trellis Coded Modulation and Viterbi soft decoding

This functional block uses BPSK in conjunction with TCM at the transmission channel modulator and Viterbi soft decoding at the transmission channel demodulator. This means the Trellis state diagram has only 2 states instead of 4 as in the QPSK model. This Trellis state diagram is given in Figure 25.

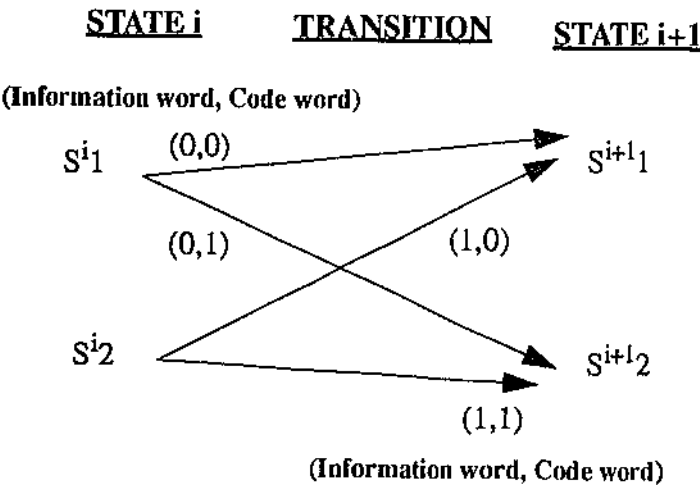


Figure 25: Trellis state diagram for BPSK

The left side of Figure 25 shows two possible encoder states with two possible state transitions from each state, that is, one information bit (two possible values). Each transition is labelled with the information word that causes this transition and the code word that is output.

The right side of Figure 25 shows the state that the encoder reaches after the corresponding transition. For instance, in state 1 with information word 0(= 0), the code word 1(= 1) is transmitted; after transition, the new state is state 2.

As before, because the Trellis encoder and Viterbi decoder models aim at TCM and coding schemes, code words are not transmitted directly. Instead, they are mapped onto complex channel symbols (modulation), which are transmitted (so are also the expected input of the decoder). The BPSK constellation mapping is different as well, as seen in Figure 26.

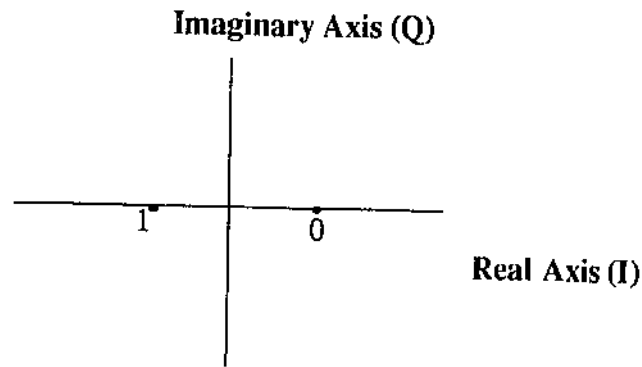


Figure 26: BPSK constellation mapping

The parameter file and COSSAP models for the BPSK channel modulation with Trellis encoding and Viterbi soft-decoding is also included in Appendix C.

3.3.2.4 DBPSK with Trellis Coded Modulation and Viterbi soft decoding

The COSSAP models for DBPSK are exactly similar to those for BPSK, except again for the

fact that in the DBPSK COSSAP models, a Differential Phase Shift Keying Modulator/Demodulator is used instead of an absolute M-ary Phase Shift Keying Modulator/Demodulator.

Please refer Appendix C for the parameter file of this COSSAP model.

3.3.3 CDMA multiplexer and demultiplexer blocks

Initially, only orthogonal 16-bits Rademacher-Walsh multiplexer and demultiplexer models were constructed. Nonetheless, these modular functional blocks can be used with other CDMA multiplexing codes such as the 7-bits Gold codes. The only changes required for these blocks to accommodate other CDMA code sets are changes to the input dataset and changes to the holding and integration ratio (*\$Walsh_Num_Bits*). Hence the following operation Walsh sequence multiplexing and demultiplexing blocks also applies to the 7-bits Gold codes. Only in the case of the 7-bits Gold codes, a ratio of 7:1 is used instead of 16:1 as with the 16-bits Rademacher-Walsh orthogonal sequences.

3.3.3.1 Walsh sequence code multiplexer and demultiplexer

As indicated in Chapter 1, we are using low correlation sequences such as the Rademacher-Walsh orthogonal sequences to multiplex the access channels in our DS-CDMA system. This multiplexing also involves a spreading of the input signal. This spreading ratio is

determined by the length of the CDMA low correlation sequences used.

The Walsh sequence multiplexer operates in a similar manner as the Barker sequence modulator. The difference between these two functional blocks lies in the Walsh sequence demultiplexer block. In the Walsh demultiplexer, the input signal is first multiplied with the expected Walsh sequence for the receiver, then the result is integrated over the length of the Walsh sequence used. No ideal sampling is required, unlike in the Barker sequence demodulator. The Walsh sequence CDMA code set used are the 16-bits Rademacher-Walsh sequences given in Chapter 1.

In the case of Gold codes, the same functional block is used but the orthogonal Walsh CDMA code set is replaced by the low correlation Gold code set. Furthermore, the ratio of holding/integration ($\$Walsh_Num_Bits$) is also changed to the bit length of the Gold code set used. The applied Gold code set is a 7-bits Gold code set.

From the developed system model, CDMA multiplexer and demultiplexer blocks for complex signals were needed. These functional block are constructed by combining two hierarchical blocks, namely two Walsh sequence multiplexers or demultiplexers. This can be achieved because the imaginary signal has the same data type as a real signal, which is a floating point number representation.

COSSAP models of the aforementioned CDMA multiplexer and demultiplexer are depicted in Appendix D.

3.3.4 Arithmetic signals adder blocks

The COSSAP libraries only have arithmetic signals adders (for integer, real and complex signals) that add up to 4 input signals. As the preliminary simulation model supports up to 16 multiple access channels, a 16 input signals adder is required for complex signals. This adder is constructed by summing the output of four 4 complex signals adders into the input of another 4 input complex signals adder. Also, as the 7-bits Gold code set can support up to a maximum of 9 simultaneous users, a 9 complex signals adder is also developed. This 9 complex signals adder is constructed by adding 3 complex signals adders into another 3 complex signals adder.

The COSSAP models of these complex signals adders are given in Appendix F.

3.4 Simulation technique for system BER evaluation

To evaluate the system's BER the standard Monte Carlo simulation method is used. This is because non-linearities within the considered DS-CDMA indoor microwave ATM LAN model made other more efficient simulation technique such as Importance Sampling unsuitable [29]. Thus the number of samples required for each simulation run is at least the reciprocal of the BER need to be observed. Indeed, for a system such as the investigated DS-CDMA indoor microwave ATM LAN model in this thesis, with BER requirement of at least 10^{-7} or lower, sample sizes of 10^7 or higher have to be gathered. Furthermore, Blum and Rosenblatt have shown in [30] that the number of samples per simulation run required to estimate a system's

true BER using standard Monte Carlo simulation method is given by

$$N > \frac{1}{e^2 P_e} \quad (30)$$

where N is the minimum number of samples required, P_e is the actual system BER and e is the normalised error given by

$$e = \frac{\text{standard deviation of } \hat{P}_e}{P_e} \quad (31)$$

Hence, for the testbed developed, in order to obtain a reasonable estimate of the true system BER at least 10 sets of independently random samples have to be gathered for each simulated system performance evaluation. The COSSAP model used to register the system BER is given in Appendix E.

4 Performance evaluation of different CDMA code sequences

4.1 Introduction

With CDMA, multiple user access is implemented utilizing user specific low correlation or orthogonal CDMA code multiplexing.

In traditional DS-CDMA systems, usually the PN sequence is also a part of the CDMA multiplexing code sequence [33]. In this thesis, the approach of separating the code-set for the PN sequence and the CDMA multiplexing code sequences has been used to provide a mean of spreading ratio control for CDMA code-set of different length.

This chapter presents a comparative evaluation of the system BER between a DS-CDMA system using traditional low correlation coding such as Gold codes and a DS-CDMA system using orthogonal coding such as Rademacher-Walsh orthogonal sequences.

4.2 Simulation System Design

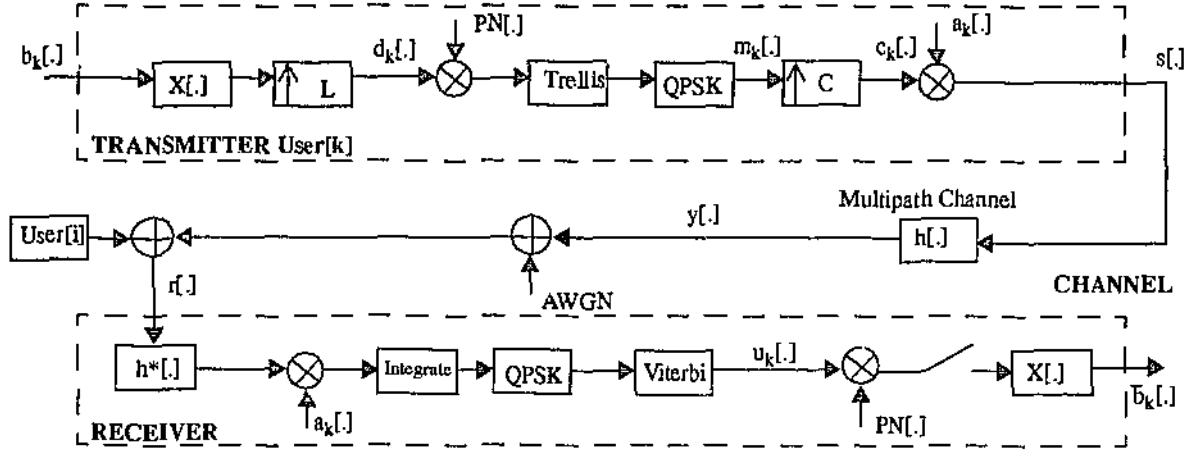


Figure 27: Block diagram of simulated DS-CDMA system

In Figure 27, a block diagram defining the DS-CDMA simulation model used to simulate the forward link of both the low correlation coded DS-CDMA system and the orthogonal coded DS-CDMA system. The binary information sequence of user k ($k = 0, 1, \dots, K-1$) is given by $b_k[.] = \dots b_k[-1], b_k[0], b_k[1], \dots$, where the samples $b_k[i] = \pm 1$ are defined at time instants iT_b and where T_b denotes the bit duration. The sequence $b_k[.]$ is first interpolated by a spreading factor of L in the manner

$$d_k[n] = \begin{cases} b_k\left[\frac{n}{L}\right]; & n \bmod L \equiv 0 \\ 0; & \text{otherwise} \end{cases}$$

which is defined at time instants nT_c , where T_c denotes the chip duration and $T_b = LT_c$.

Here and hereafter, i and n denote the bit and the chip numbers, respectively, and refer to the

defined time instants. The interpolated information sequence $d_k[\cdot]$ of user k is then multiplied with a common spreading code, $PN[\cdot]$ (the 11-bits Barker sequence is used in this model [1]) then this wideband information signal is modulated via a QPSK modulation with TCM, producing a stream of wideband transmission symbols, $m_k[\cdot]$. Finally, this wideband transmission symbol is further interpolated by a factor of C , giving the already wideband transmission a further spread. The resultant signal, $c_k[\cdot]$, is then multiplied with the user specific CDMA code $a_k[\cdot]$ in order to form the transmitted sequence $s[\cdot]$.¹ The output sequence $y[\cdot]$ of the noiseless multipath channel² $h[\cdot]$ is distorted by an additive white Gaussian noise sequence $z[\cdot]$ with zero mean and variance $N_0/2$ (N_0 is the one-sided noise spectra density). The received sequence $r_k[\cdot]$ is first filtered by a channel matched filter $h^*[\cdot]$ ³, then multiplied with the user specific code $a_k[\cdot]$ and after that, integrated over the duration of C bits. Note that selective channel multiplexing using CDMA code sequences in this system is performed at the transmitting and receiving ends of both the transmitter and receiver to increase the robustness of the DS-CDMA system to other system interferences (i.e.

-
1. Note that the separation of PN spreading and CDMA code spreading allows for a control of the final spread ratio of the signal. Usually the CDMA code spread is fixed to the period length of the CDMA code, thus for a fixed transmission bandwidth, the ratio of spreading by the PN sequence has to be adjusted accordingly.
 2. In this chapter only the additive white Gaussian noise channel is evaluated. Current work is being carried out to evaluate system performance in multipath channels such as the Rayleigh fading channel and the Rician fading channel.
 3. Only in simulation with multipath channels as indicated in Footnote 2.

noise and other access channels). After selective demultiplexing, the received signal is further demodulated using a QPSK demodulator with Viterbi soft decoding. The output signal, $u_k[\cdot]$ is multiplied with the general spreading sequence $PN[\cdot]$ and sampled every L -th *chip* in order to get $\hat{b}_k[\cdot]$ as an estimated of $b_k[\cdot]$.

In this chapter, the pulse shaping $X[\cdot]$ is chosen as a rectangular function of length L *chips* and unit amplitude.

4.2.1 Simulating varying distance transmitter

To simulate the effects of varying distance of transmitter from the receiver, each transmitter in the simulated DS-CDMA model transmits at a signal power of

$$\text{Signal Power} = \text{Noise Power} \times 10^{\frac{k}{1000}} \text{ mW}$$

For the evaluation of BER performance, this chapter only looks at the worst case scenario, which is the BER observed at access channel $k = 0$. (Refer Figure 30 for the COSSAP layout of the simulation model.)

4.3 CDMA coding

Traditional CDMA systems uses low correlational code sequences such as Gold codes and Kasami codes for CDMA code multiplexing [31,33]. In recent developments, such as in Qualcomm's proposed CDMA cellular system [34], orthogonal coding is used for CDMA code multiplexing. This chapter presents a performance evaluation of both systems in terms of BER for the worst case channel in a forward-link network configuration.

4.3.1 Optimum Gold code set

For the purpose of this chapter, an optimum set of Gold code sequences as proposed in [31] has been selected. Derivation of the preferred pairs maximum-length sequences which make up this code set can be found in [31].

The Gold set used is 7-bits in length and can accommodates up to 9 users. Each user and their specific CDMA 7-bits Gold code is listed in Table 10.

| User, k | 7-bits Gold code sequence |
|-----------|---------------------------|
| 0 | 1 -1 1 1 1 -1 -1 |
| 1 | 1 1 -1 1 1 -1 1 |
| 2 | -1 1 1 -1 1 -1 1 |
| 3 | -1 1 -1 1 1 -1 -1 |
| 4 | 1 1 -1 -1 1 1 -1 |
| 5 | 1 -1 -1 -1 -1 -1 1 |

Table 10: User specific CDMA 7-bits Gold code set

| User, k | 7-bits Gold code sequence |
|-----------|---------------------------|
| 6 | -1 -1 1 -1 -1 1 -1 |
| 7 | 1 1 1 1 -1 1 1 |
| 8 | -1 -1 -1 1 1 1 1 |

Table 10: User specific CDMA 7-bits Gold code set

Cross correlation between 7 bits Gold code sequence set

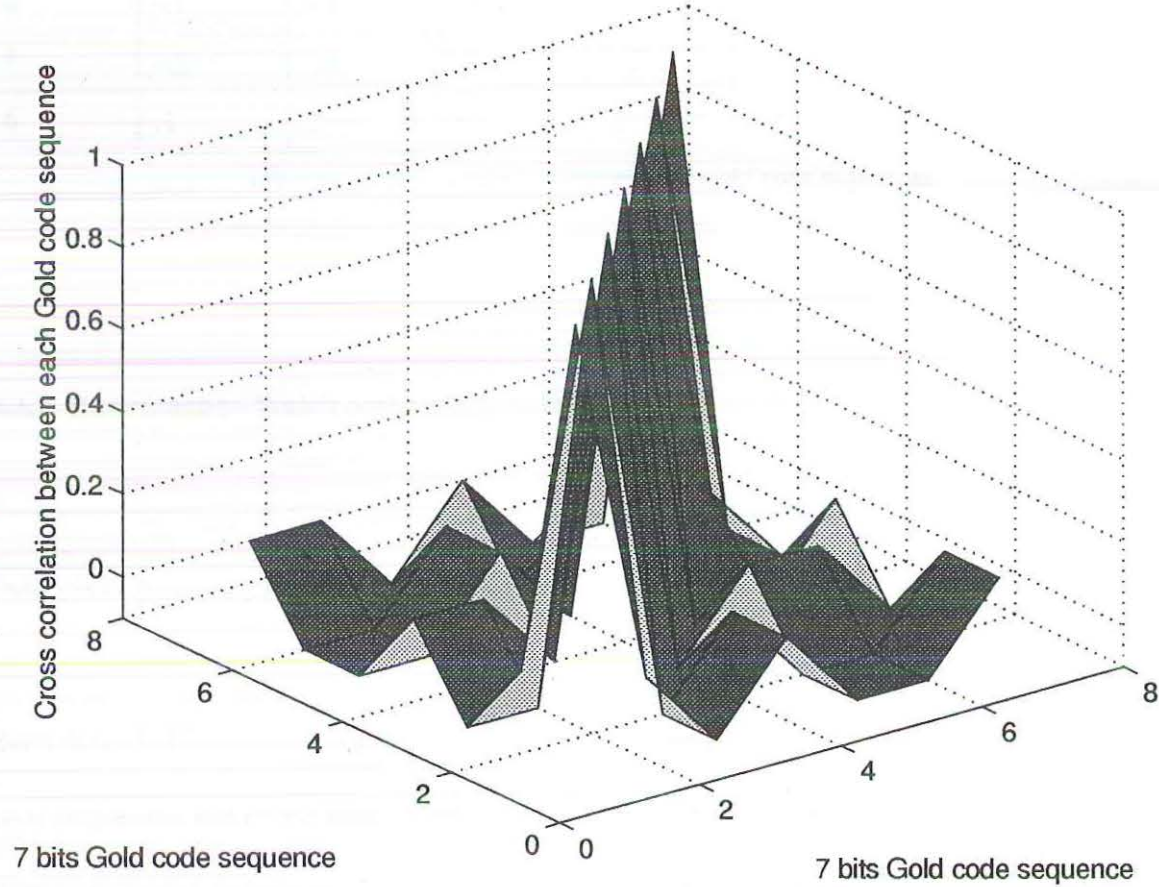


Figure 28: Cross-correlation of the 7-bits Gold code sequences

Figure 28 shows the cross-correlation of the 7-bits Gold code sequences. As indicated in Table

11, the cross-correlation between each of the 7-bits Gold code sequence and another is ± 0.1 .

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 1 | .1 | -.1 | .1 | -.1 | -.1 | .1 |
| 1 | .1 | 1 | -.1 | .1 | -.1 | -.1 | .1 |
| 2 | -.1 | -.1 | 1 | -.1 | .1 | .1 | -.1 |
| 3 | .1 | .1 | -.1 | 1 | -.1 | -.1 | .1 |
| 4 | -.1 | -.1 | .1 | -.1 | 1 | .1 | -.1 |
| 5 | -.1 | -.1 | .1 | -.1 | .1 | 1 | -.1 |
| 6 | .1 | .1 | -.1 | .1 | -.1 | -.1 | 1 |

Table 11: Cross-correlation between the Gold code sequences

4.3.2 Rademacher-Walsh orthogonal code sequences

As for the orthogonal CDMA code set, an optimum orthogonal 16-bits Rademacher-Walsh code set as proposed in Chapter 1 is used.

Even though this optimal orthogonal code set can support up to 16 simultaneous users, only 9 code sequences out of the total 16 are used in the simulation model. This is to ensure an equivalent comparability between the two evaluated systems. Each user and their respective orthogonal 16-bits Rademacher-Walsh code sequence are listed in Table 12.

| User k | 16-bits Rademacher-Walsh sequence |
|----------|---|
| 0 | 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 1 | -1 -1 -1 -1 -1 -1 -1 -1 1 1 1 1 1 1 1 1 |
| 2 | -1 -1 -1 -1 1 1 1 1 1 1 1 1 -1 -1 -1 -1 |
| 3 | 1 1 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 |
| 4 | 1 1 -1 -1 -1 -1 1 1 1 1 -1 -1 -1 -1 1 1 |
| 5 | -1 -1 1 1 1 1 -1 -1 1 1 -1 -1 -1 -1 1 1 |
| 6 | -1 -1 1 1 -1 -1 1 1 1 1 -1 -1 1 1 -1 -1 |
| 7 | 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 |
| 8 | 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 1 -1 -1 1 |

Table 12: User specific CDMA orthogonal 16-bits Rademacher-Walsh code set

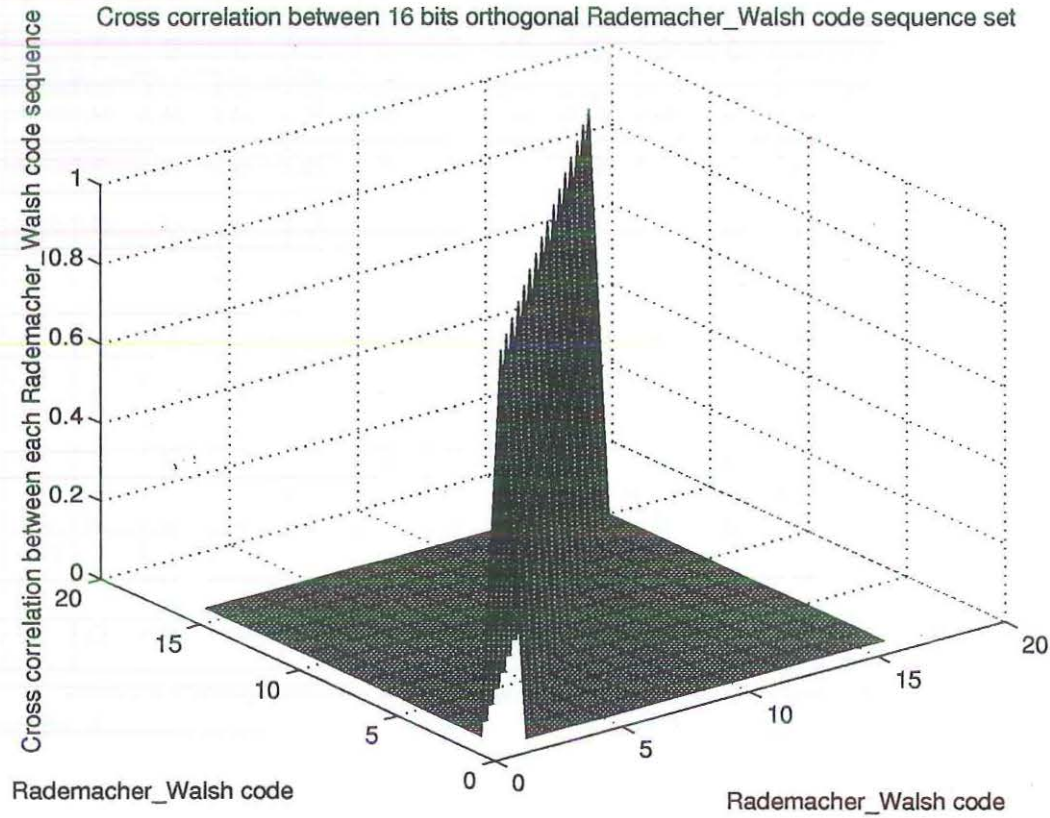


Figure 29: Cross-correlation of the 16-bits orthogonal Rademacher-Walsh code sequences

Figure 29 shows the cross-correlation of all 16 orthogonal 16-bits Rademacher-Walsh code sequences. From Table 13, the orthogonal nature of the 16-bits Rademacher-Walsh code sequences is observed in the 0 cross-correlation between each 16-bits Rademacher-Walsh sequence with another.

| k | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-----|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |

Table 13: Cross-correlation between the 16-bits Rademacher-Walsh code sequences

4.4 Simulation method

BER of the simulation model is evaluated in the COSSAP environment. Estimate of the BER is derived using Monte Carlo simulation method for a sample size of 10^7 samples for each SNR. At least 10 errors is registered for each Monte Carlo BER estimation. The COSSAP simulation model used is given in Figure 30.

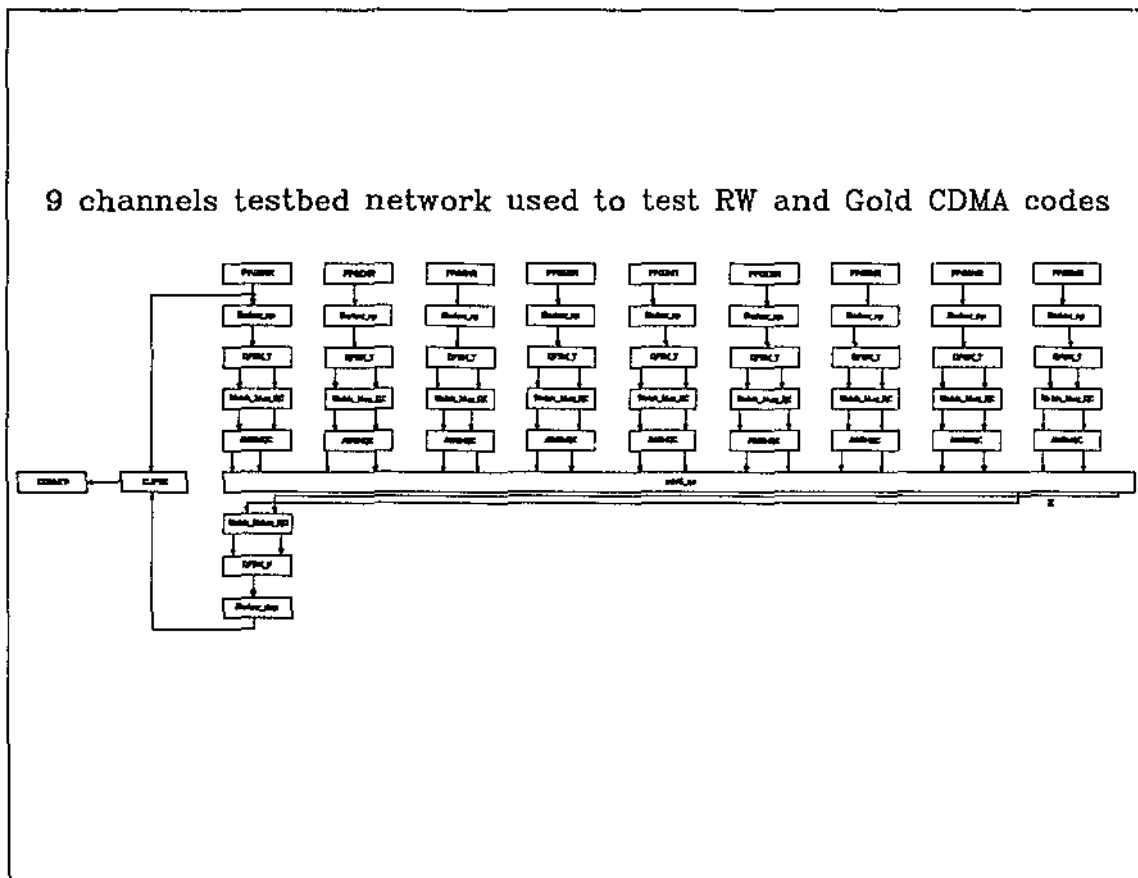


Figure 30: COSSAP model of 9 channels DS-CDMA system simulation

4.5 Simulation results

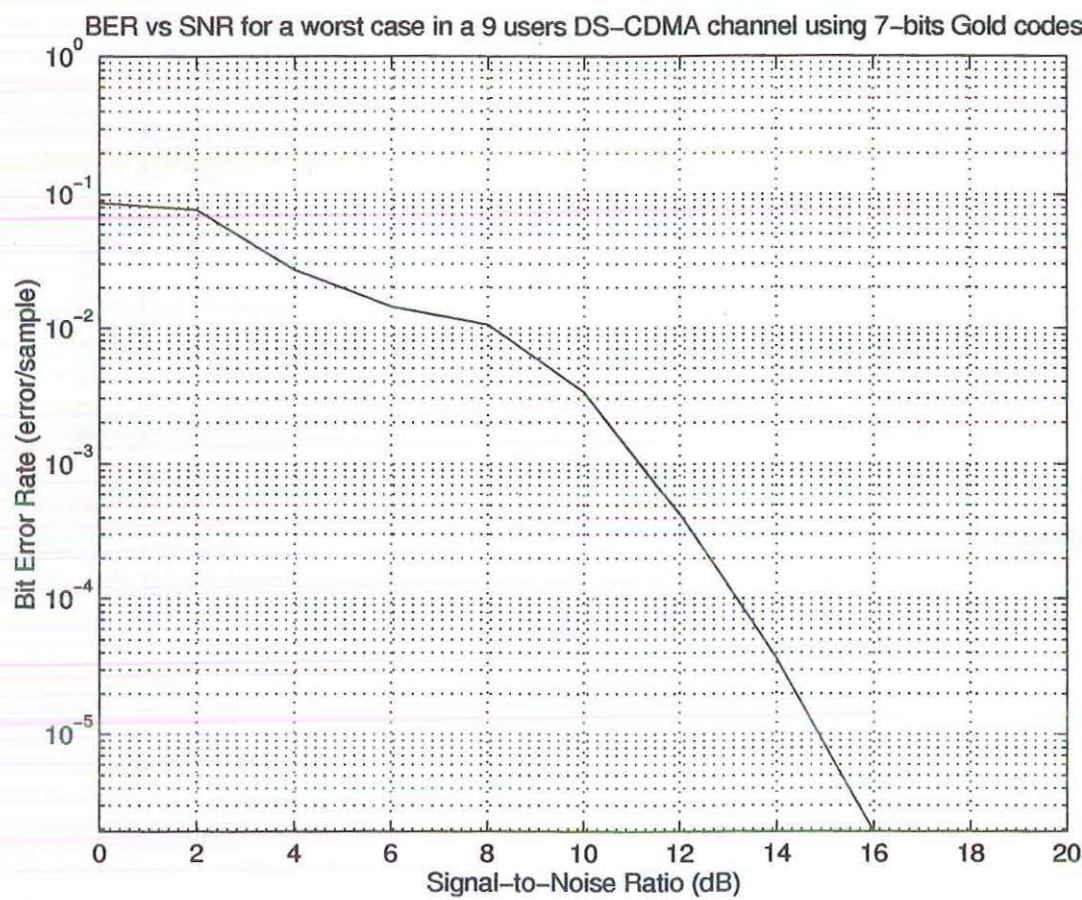


Figure 31: BER of the DS-CDMA system using 7-bits Gold code sequences

Figure 31 presents the BER performance of the DS-CDMA system utilizing low correlational 7-bits Gold code sequences. The SNR of interest for our DS-CDMA indoor microwave ATM LAN is from 0 to 20 dB.

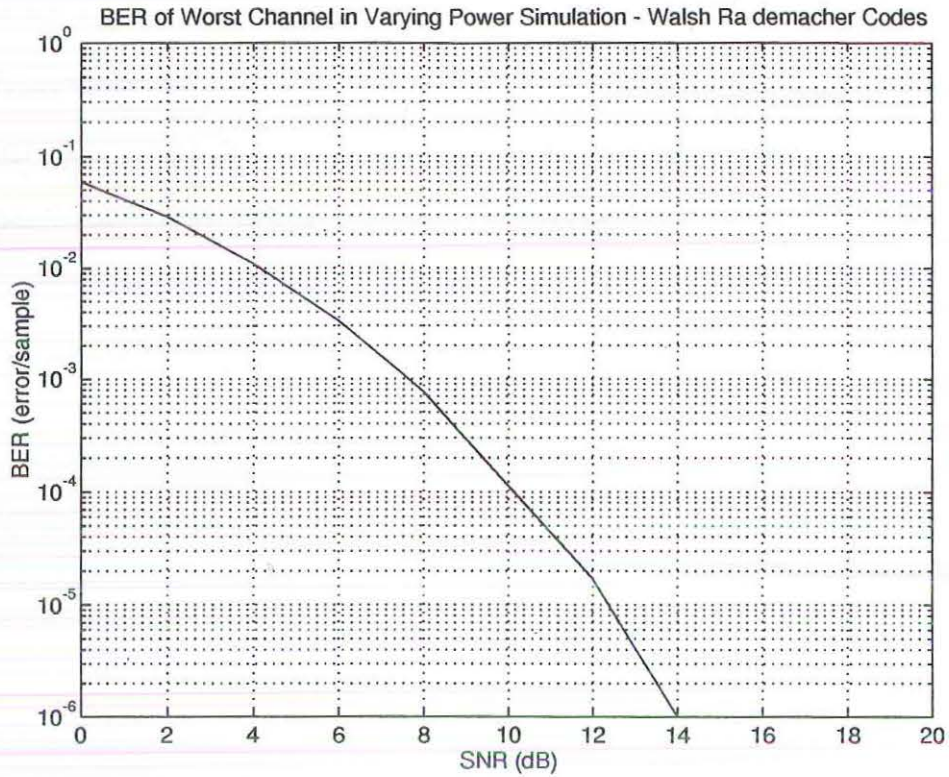


Figure 32: BER of DS-CDMA system using the 16-bits Rademacher-Walsh sequences

Figure 32 presents the BER performance of the same DS-CDMA system utilizing the 16-bits orthogonal Rademacher-Walsh sequences.

From these preliminary results, it can be said that orthogonal codes do have a performance advantage over low correlation code-sets such as the 7-bits Gold codes. On the other hand, the number of low correlation codes in a set is not fixed at the bit length of the code sequence used, while with orthogonal sequences such as the Walsh-Rademacher codes, the number of code sequences is fixed at the bit length of the sequence being used.

Hence, the low correlation 7-bits Gold code set can accommodate $T_{Gold} + 2$ number of simultaneous users, but the orthogonal code set can only support T_{RW} simultaneous users. Where T_{Gold} = bits length of Gold code; T_{RW} = bits length of the Rademacher-Walsh orthogonal code. Where the orthogonal codes gained in system efficiency it lost in system throughput in comparison to low correlation Gold codes.

5 Conclusion

This thesis has presented a novel design for a simulation model of DS-CDMA indoor microwave ATM LAN. The developed simulation model is implemented in COSSAP and is highly modular and flexible to parameter changes. Therefore it may be used in the performance evaluation of different physical layer components for such a network. An example of such a contribution is given in Chapter 4.

Currently the model developed in this thesis is being used by the 'Wireless ATM Hub Design' team within the Cooperative Research Centre for Broadband Telecommunication and Networking in their design work for a wireless ATM hub.

The work done in this thesis is original and has been accepted for publication at national and international conferences [9, 36, 37].

Items for further research originating from the work done in this thesis are as follow:

- Performance evaluation of different physical layer components of DS-CDMA indoor microwave ATM LAN. This includes evaluation of different:
 - PN spreading sequences
 - Other CDMA coding schemes
 - Different channel modelling and performance of system in these channels
 - Power control
 - Different channel modulation schemes
 - Error Correction Coding

-
- Evaluation of hybrid spread spectrum systems
 - Increase simulation efficiency for a large scale simulation using statistical and mathematical modelling techniques.

6 References

- [1] Struhsaker, P., "Draft Standard Text for Direct Sequence Spread Spectrum Medium Dependent PHY Sublayer", 1994, Doc:IEEE 802.11
- [2] Diepstraten, Wim, "IEEE 802.11 Wireless Access Method and Physical Layer Specifications: Wireless Network Performance Modelling Approach", February, 1992, DOC:IEEE P802.11-92/26.
- [3] Elhakeem, Ahmed K., Hafez, H. M. & Mahmoud, Samy A., "Spread Spectrum Access to Mixed Voice-Data Local Area Networks", IEEE Journ. Selected Areas Comms., VOL. SAC-1, NO. 6, December 1983, pp 1054-1069.
- [4] Lee, J. R. & Un, C. K., "A Code-Division Multiple-Access Local Area Network", IEEE Trans. Comms., VOL. COM-35, NO. 6, June 1987, pp 667-671.
- [5] Aghvami, A., Robertson, I. D. & Mohamed, S. A., "High Bit-Rate Indoor Radio Communications", Proceedings of the Third IEE Conference on Telecommunications, NO. 331, pp 101-106.
- [6] Marshall, A., Spracklen, C.T. & Smythe, C., "An Experimental Testbed for a Direct Sequence Spread Spectrum Local Area Network", Second IEE National Conference on Telecommunications, NO. 300, pp 288-293.
- [7] Karkkainen, Kari H. A., "CODSIM: A Simulation Environment for Spreading Codes", VTC 1994 'Creating Tomorrow's Mobile System', Proceedings of IEEE 44th Vehicular Technology Conference, VOL. 1, 1994, pp 535-539.
- [8] Tan, Keng T. & Wysocki, Tadeusz, "Implementing Multiple Access In High Data Rate Mobile Network Using DS-CDMA", Proceedings of Australian Telecommunication Networks and Applications Conference 1995, December 1995, Sydney - *to be published*.
- [9] Tan, Keng T. & Wysocki, Tadeusz, "Simulation Model of DS-CDMA Indoor Microwave ATM LAN - Theoretical Foundation", February 1995, Australian Telecommunication Research Institute, Radio Signal Laboratory Internal Technical Report Series, ATRI-RSL-TR-001.
- [10] Pickholtz, Raymond L., Schilling, Donald L., Milstein, Laurence B., "Theory of Spread-Spectrum Communications - A Tutorial", May 1982, IEEE Trans. Comm., vol. COM-30, pp. 855-884.
- [11] Dixon, Robert C., "Spread Spectrum Systems with Commercial Applications", Third Edition., 1994, Wiley-Interscience Publication:New York.

-
- [12] Lee, Edward A. & Messerschmitt, David G., "Digital Communication", Second Edition., 1994, Kluwer Academic Publishers: USA.
- [13] Anonymous, "An Overview of the application of Code Division Multiple Access (CDMA) to Digital Cellular Systems and Personal Cellular Networks", May 1992, QUALCOMM Inc.:USA.
- [14] Golomb, Solomon W., "Shift Register Sequences", 1967, Holden-Day Inc.:San Francisco.
- [15] Lee, William C. Y., "Mobile Communications Design Fundamentals", Second Edition., 1993, Wiley Series In Telecommunications:New York.
- [16] Lange, F. H., "Correlation Techniques: Foundations and Applications of Correlation Analysis in Modern Communications, Measurement and Control", Second Edition, 1967, Iliffe Books Ltd.:London.
- [17] Beauchamp, K. G., "Walsh Functions and their applications", 1975, Academic Press:London.
- [18] Harmuth, Henning F., "Transmission of Information by Orthogonal Functions", Second Printing, 1970, Springer-Verlag:Berlin.
- [19] Magill, D.T. & Kirke, T.G., "Optimal Short-range CDMA Codes of Length 16 for Wireless Communications", Proceedings of the 1992 IEEE International Conference on Selected Topics in Wireless Communications, pp 195-198.
- [20] Besslich, P.W., "Walsh Function Generators for Minimum Orthogonality Error", November, 1973, IEEE Trans. EMC, vol. 15 no. 4, pp. 177-180.
- [21] Honcharenko, W., Bertoni, H. L. & Dailing, J., "Mechanism Governing Wireless Propagation Between Different Floors in Buildings", 1993, IEEE Trans. on Antennas and Propagation, Vol. 41, No. 6, p. 787.
- [22] Linnartz, J. P., "Narrowband Land-Mobile radio Networks", 1993, Artech House Inc.:MA.
- [23] Schwartz, M., Bennett, W. R. & Stein, S., "Communication Systems and Techniques", 1966, McGraw-Hill Book Company:New York.
- [24] Taub, H. & Schilling, D. L., "Principles of Communication Systems", First Edition, 1971, McGraw-Hill Book Company:New York.
- [25] Anonymous, "COSSAP Overview and User Guide. Version 6.7", December, 1994, Synopsys: CA.

-
- [26] Anonymous, "COSSAP Stream Driven Simulator User Guide. Version 6.7", December, 1994, Synopsys: CA.
- [27] Anonymous, "COSSAP File Organisation and Formats. Version 6.7", December, 1994, Synopsys: CA.
- [28] Luise, Marco, "A Survey of CDMA Techniques for Modern and Personal Satellite Communication Networks", March 27th 1995, Italy - Australia Scientific Weeks 1995: Sydney.
- [29] Jung, Pierre & Ruprecht, Jurg, "Importance Sampling Technique for Multi-User CDMA-Like Systems in Discrete-Time Multipath Fading Environments", Proceedings of the IEEE Global Telecommunications Conference - GLOBECOMM '93 Houston, VOL. 1, pp 59-63.
- [30] Blum, J. R. & Rosenblatt, J. I., "Probability and Statistics", 1972, Saunders: Philadelphia, PA.
- [31] Rothaus, Oscar S., "Modified Gold Codes", IEEE Trans. on Info. Theory, VOL. 39, NO. 2, March 1993, pp 654-656.
- [32] Yuan, Chengchi & Komo, John J., "Error Correcting CDMA Communication System", Proceedings of the Twenty Second Southwestern Symposium On System Theory, pp 215-218.
- [33] Karkkainen, K. H. A., Laukkanen, M. J. & Tarnanen, H. K., "Performance of asynchronous DS-CDMA system with long and short spreading codes", IEEE Elec. Letters, June 1994, VOL. 30, NO. 13, pp 1035-1036.
- [34] Komo, John J. & Liu, Shyh-Chang, "Modified Kasami Sequences for CDMA", Proceedings of the Twenty Second Southwestern Symposium On System Theory, pp 219-222.
- [35] Anonymous, "Wideband Spread Spectrum Digital Cellular System Dual-Mode Mobile Station-Base Station Compatibility Standard", May 1992, QUALCOMM Inc.:USA.
- [36] Tan, Keng T. & Wysocki, Tadeusz, "COSSAP Simulation Model of DS-CDMA Indoor Microwave ATM LAN", Proceedings of Workshop on Applications of Radio Science, Canberra, June 1995.
- [37] Tan, Keng T. & Wysocki, Tadeusz, "A Software Simulation Testbed for a Direct Sequence Code Division Multiple Access Asynchronous Transfer Mode Wireless Local Area Network in COSSAP", Proceedings of IEEE INFOCOMM, March 1996, San Francisco - *submitted for publication*.

Appendix A: Source, Sink and Matched Filter functional blocks

Following are the configuration files and COSSAP models for the **Real Signal Source Generator** (*PPGENR*), **BER Counter** and **0.5 Raised-Cosine Roll-off Matched Filter** (*rcrfqc*).

A.1 Real Signal Source Generator (*PPGENR*)

| | |
|--|---|
| ***** | |
| SYSTEM CONFIGURATION | |
| ***** | |
| | |
| layout name | : unknown |
| user name | : Alfred Tan at python |
| installation id | : 204 |
| current date | : 09-oct-95 13:28 |
| creation date | : unknown |
| used libraries | : 0.0.0 \${COSSAP_MODELS}/mod63/mod63.mdb |
| number of models | : 1 |
| assignment file | : undefined |
| | |
| ----- | |
| | |
| BLOCK : | |
| model name | : ppgenr |
| model id | : 0.0.0.12 |
| sequence number | : 4 |
| | |
| INPUT_DATASETS : | |
| 1 PATTERN | |
| dataset id | : Input1 |
| file type | : implicit |
| file format | : COSSAP_BINARY |
| filename | : \${COSSAP_USER_C}\Input1.txt |
| | |
| OUTPUT_PORTS : | |
| 1 (R) OUTPUT1 |SIG_0003 |
| output_rate : default | |
| | |
| SELECTED IMPLEMENTATION : | |
| implementation..... = DEFAULT_IMPLEMENTATION | |

Table 14: Configuration file for COSSAP model *ppgenr*

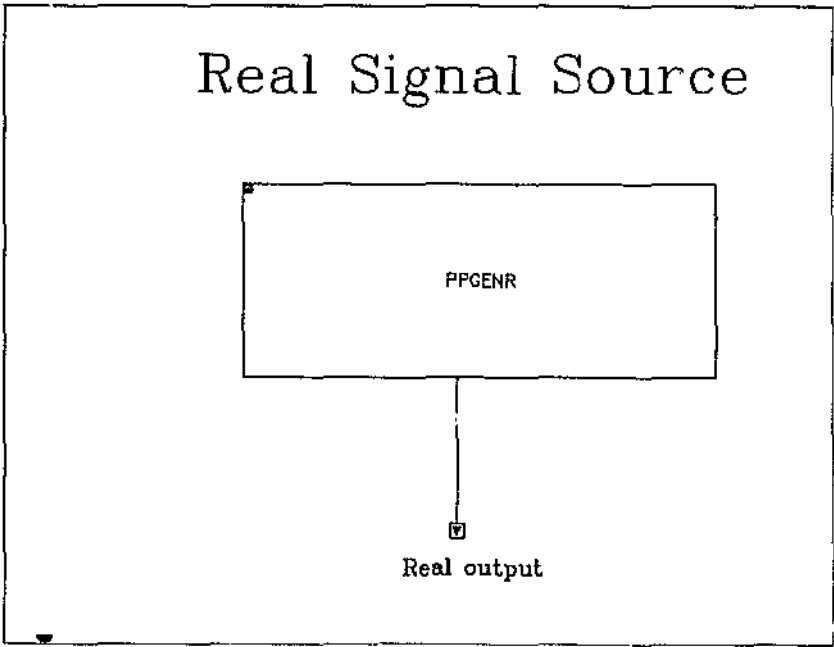


Figure 33: COSSAP model (*ppgenr*)

| | | |
|---|--------|--|
| # | 16 (R) | |
| | 0 | |
| | 0 | |
| | 0 | |
| | 0 | |
| | 0 | |
| | 1 | |
| | 0 | |
| | 1 | |
| | 1 | |
| | 1 | |
| | 0 | |
| | 0 | |
| | 1 | |
| | 1 | |
| | 1 | |
| | 1 | |
| # | -1 (R) | |

Table 15: A sample of input data file for *ppgenr*

A.2 BER Counter

| | |
|--------------------------|---|
| ***** | |
| SYSTEM CONFIGURATION | |
| ***** | |
| | |
| layout name | : BER Counter |
| user name | : Alfred Tan at python |
| installation id | : 204 |
| current date | : 09-oct-95 13:24 |
| creation date | : unknown |
| used libraries | : 0.0.0 \${COSSAP_MODELS}/mod63/mod63.mdb |
| | 0.0.2 \${COSSAP_MODELS}/dsp/dsp.mdb |
| number of models | : 2 |
| assignment file | : undefined |
| | |
| ----- | |
| | |
| BLOCK : | |
| model name | : countp |
| model id | : 0.0.0.237 |
| sequence number | : 5 |
| | |
| PARAMETERS : | |
| 1 (I) MAX_NUM_ONES..... | undefined |
| 2 (I) MAX_NUM_ITEMS..... | undefined |
| 3 (R) X_VALUE..... | undefined |
| | |
| OUTPUT_DATASETS : | |
| 1 COUNTS | |
| dataset id | : |
| chart type | : LINE |
| plot format | : RATE |
| file type | : implicit |
| file format | : COSSAP_BINARY |
| filename | : (disabled) |
| | |
| INPUT_PORTS : | |

Table 16: Configuration file for the COSSAP model of BER Counter

| |
|--|
| 1 (I) IN_BIN..... connected to : SIG_0003 |
| scheduling : self-scheduled |
| SELECTED IMPLEMENTATION : |
| implementation..... = DEFAULT_IMPLEMENTATION |
| ----- |
| BLOCK : |
| model name : cmpne |
| model id : 0.0.2.32 |
| sequence number : 4 |
| INPUT_PORTS : |
| 1 (R) Input1..... connected to : SIG_0004 |
| scheduling : self-scheduled |
| 2 (R) Input2..... connected to : SIG_0005 |
| scheduling : self-scheduled |
| OUTPUT_PORTS : |
| 1 (I) Output..... connected to : SIG_0003 |
| output_rate : default |
| SELECTED IMPLEMENTATION : |
| implementation..... = DEFAULT_IMPLEMENTATION |

Table 16: Configuration file for the COSSAP model of BER Counter

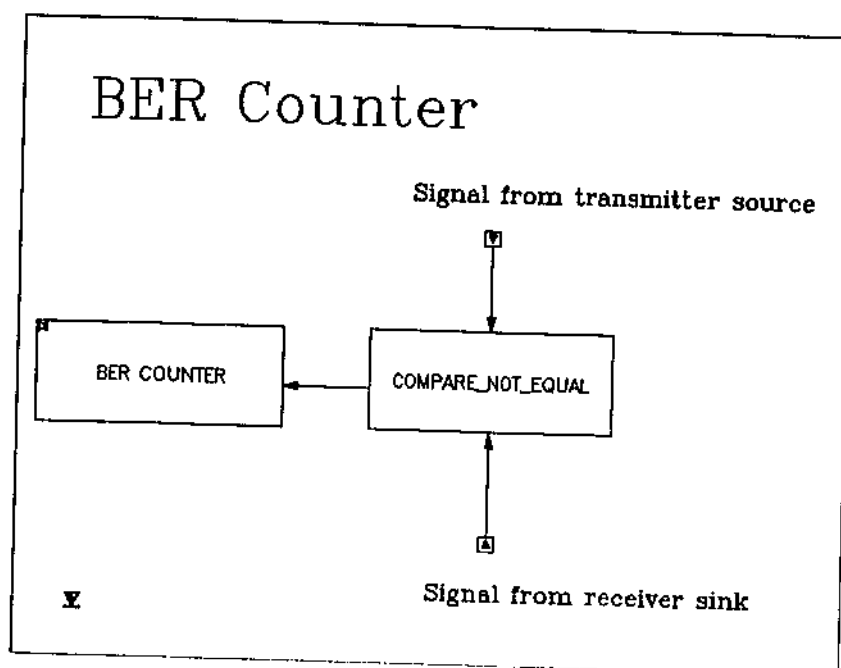


Figure 34: COSSAP model of the BER counter

A.3 0.5 Raised-Cosine Roll-off Matched Filter (*rcrfqc*)

| | |
|---------------------------------|---|
| ***** | |
| SYSTEM CONFIGURATION | |
| ***** | |
| | |
| layout name | : rcrfqc |
| user name | : Alfred Tan at python |
| installation id | : 204 |
| current date | : 10-oct-95 13:47 |
| creation date | : unknown |
| used libraries | : 0.0.0 \${COSSAP_MODELS}/mod63/mod63.mdb |
| number of models | : 1 |
| assignment file | : undefined |
| | |
| ----- | |
| | |
| BLOCK : | |
| model name | : rcrfqc |
| model id | : 0.0.0.158 |
| sequence number | : 4 |
| | |
| PARAMETERS : | |
| 1 (R) SAMPLING_TIME..... | 1.0 |
| 2 (R) CUTOFF_FREQUENCY_3DB..... | 0.4 |

Table 17: Configuration file for COSSAP model *rcrfqc*

| | | |
|----------------------------|-----------------------------|------|
| 3 (R) ROLLOFF..... | = | 0.5 |
| 4 (R) GAIN..... | | 1.0 |
| 5 (I) FFT_LENGTH..... | | 2048 |
| 6 (R) ATTENUATION_ISI..... | | 0.0 |
| 7 (R) ATTENUATION_NCD..... | | 0.0 |
| | | |
| INPUT_PORTS : | | |
| 1 (R) IN_REAL..... | connected to : SIG_0003 | |
| | scheduling : self-scheduled | |
| 2 (R) IN_IMAG..... | connected to : SIG_0004 | |
| | scheduling : self-scheduled | |
| | | |
| OUTPUT_PORTS : | | |
| 1 (R) OUT_REAL..... | connected to : SIG_0005 | |
| | output_rate : default | |
| 2 (R) OUT_IMAG..... | connected to : SIG_0006 | |
| | output_rate : default | |
| | | |
| SELECTED IMPLEMENTATION : | | |
| implementation..... | = DEFAULT_IMPLEMENTATION | |

Table 17: Configuration file for COSSAP model *rcrfqc*

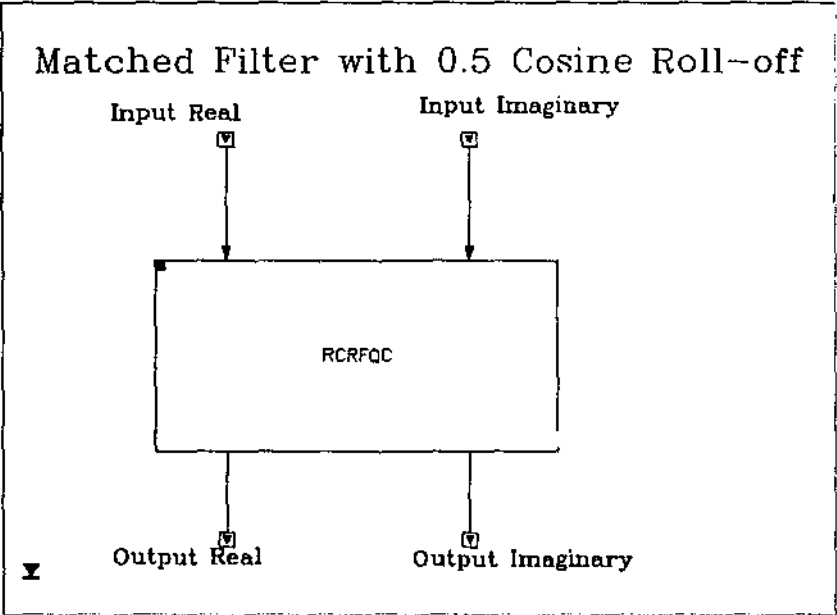


Figure 35: COSSAP model (*rcrfqc*)

Appendix B: Pseudorandom Noise spreading and despreading functional blocks

Following are the configuration files and COSSAP models for the 11-bits Barker Sequence Spreading Functional Block (*Barker_sp*) and the 11-bits Barker Sequence Despreading Functional Block (*Barker_dsp*).

B.1 11-bits Barker Sequence Spreading Functional Block (*Barker_sp*)

| |
|-----------------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| Barker_sp |
| |
| PARAMETERS : |
| (I) \$B_Hold |
| used in block HOLDR (4) : |
| HOLD_FACTOR..... = \$B_Hold |
| |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |

Table 18: Configuration file for COSSAP model *Barker_sp*

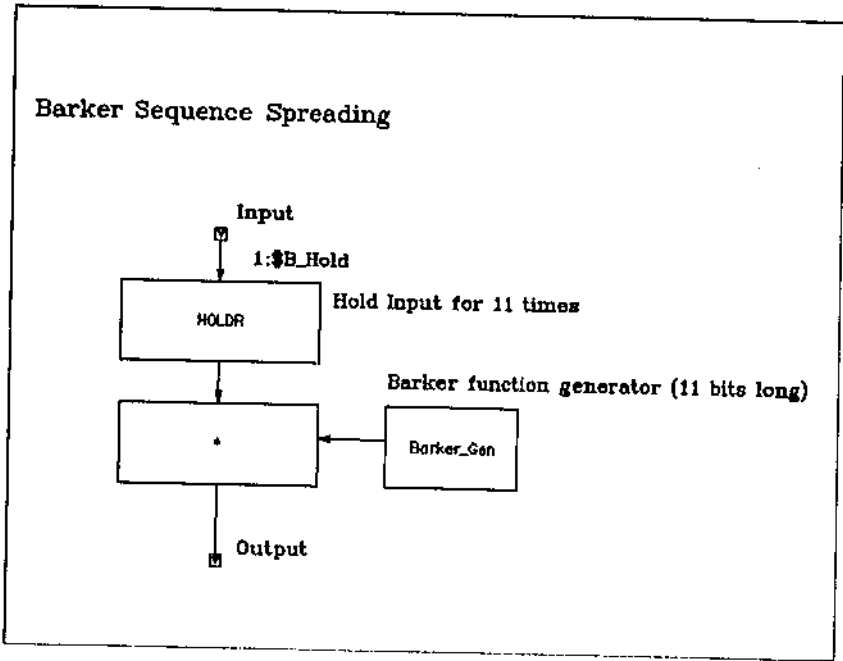


Figure 36: COSSAP model (*Barker_sp*)

B.2 11-bits Barker Sequence Despreading Functional Block (*Barker_dsp*)

| |
|------------------------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| Barker_dsp |
| |
| PARAMETERS : |
| (I) \$B_Hold1 |
| used in block SRDR (7) : |
| DECIMATION_FACTOR..... = \$B_Hold1 |
| |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |

Table 19: Configuration file for COSSAP model *Barker_dsp*

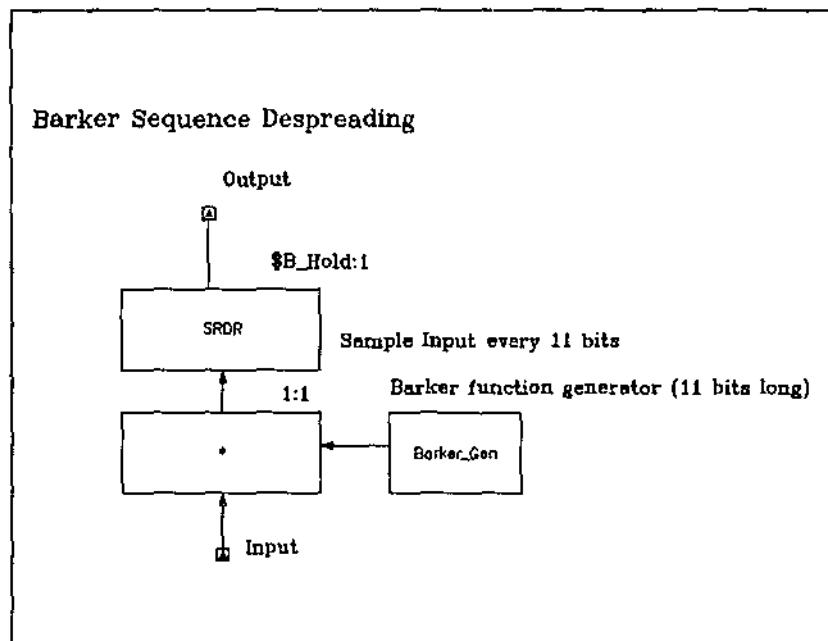


Figure 37: COSSAP model (*Barker_dsp*)

| # | 11 (R) |
|---|--------|
| | 1 |
| | -1 |
| | 1 |
| | 1 |
| | -1 |
| | 1 |
| | 1 |
| | 1 |
| | -1 |
| | -1 |
| | -1 |
| # | -1 (R) |

Table 20: 11-bits Barker sequence input data file

Appendix C: Transmission channel modulator and demodulator blocks

Following are the configuration files and COSSAP models for:

- the BPSK Modulator (*BPSK_T*),
- the BPSK Demodulator (*BPSK_V*),
- the DBPSK Modulator (*DBPSK_T*),
- the DBPSK Demodulator (*DBPSK_V*),
- the QPSK Modulator (*QPSK_T*),
- the QPSK Demodulator (*QPSK_V*),
- the DQPSK Modulator (*DQPSK_T*)
- the DQPSK Demodulator (*DQPSK_V*)

All the transmission channel modulation models have TCM and all the transmission channel demodulation models use Viterbi soft decoding. The Trellis/Viterbi encoding/decoding file for each of the functional model is also presented.

C.1 BPSK Modulator (*BPSK_T*)

| |
|---------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| MODEL NAME : |
| BPSK_T |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |

Table 21: Configuration file for COSSAP model *BPSK_T*

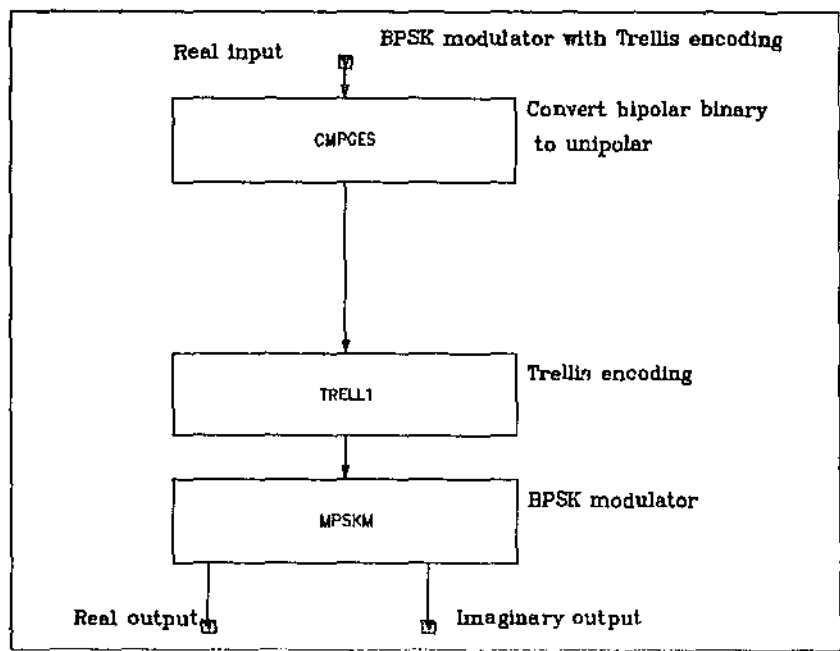


Figure 38: COSSAP model (*BPSK_T*)

C.2 BPSK Demodulator (*BPSK_V*)

| |
|--|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| BPSK_V |
| |
| PARAMETERS : |
| (I) \$SURVIVOR_DEPTH |
| used in block VITRB1 (4) : |
| SURVIVOR_DEPTH..... = \$SURVIVOR_DEPTH |
| |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| (R) \$INPUT_IMAG |
| |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |

Table 22: Configuration file for COSSAP model *BPSK_V*

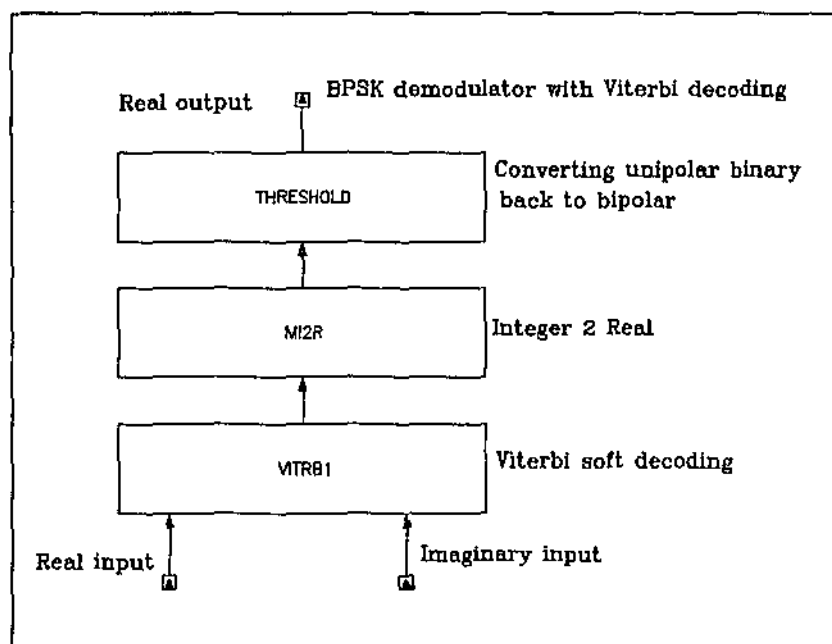


Figure 39: COSSAP model (BPSK_V)

| |
|--|
| .NUMBER_OF_STATES |
| 2 |
| .MAXIMUM_#_OF_PARALLEL_BRANCHES_P |
| 1 |
| .NUMBER_OF_BRANCHES_JOINING_AT_ONE_STATE |
| 2 |
| .NUMBER_M_OF_CHANNEL_SYMBOLS |
| 2 |
| .SELECT_MODULATION_(MPSK=0_QAM=1) |
| 0 |
| .WITH_DIFFERENTIAL_CODING_(YES=1_NO=0) |
| 0 |
| *.TRELLIS_TABLE_(STATE,CHANNEL_SYMBOL)" |
| ----- ----- ---- ---- ---- ---- |
| 1 1 0 |
| 1 2 1 |
| 2 1 1 |
| 2 2 0 |
| .COMPONENT_TABLE_FOR_CHANNEL_SYMBOLS |
| ----- ----- ----- ----- |
| 0 1 0 |
| 1 -1 0 |
| .END |

Table 23: Trellis/Viterbi encoding/decoding parameter file for COSSAP model *BPSK_T* & *BPSK_V*

C.3 DBPSK Modulator (*DBPSK_T*)

| |
|---------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| DBPSK_T |
| |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |

Table 24: Configuration file for COSSAP model *DBPSK_T*

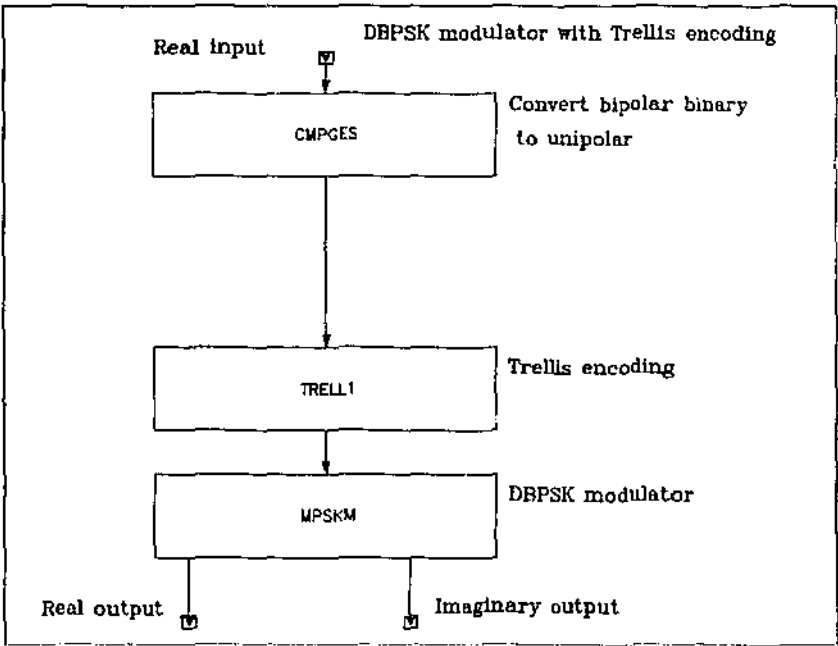


Figure 40: COSSAP model (*DBPSK_T*)

C.4 DBPSK Demodulator (DBPSK_V)

| |
|--|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| MODEL NAME : |
| DBPSK_V |
| PARAMETERS : |
| (I) \$SURVIVOR_DEPTH |
| used in block VITRB1 (4) : |
| SURVIVOR_DEPTH..... = \$SURVIVOR_DEPTH |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| (R) \$INPUT_IMAG |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |

Table 25: Configuration file for COSSAP model DBPSK_V

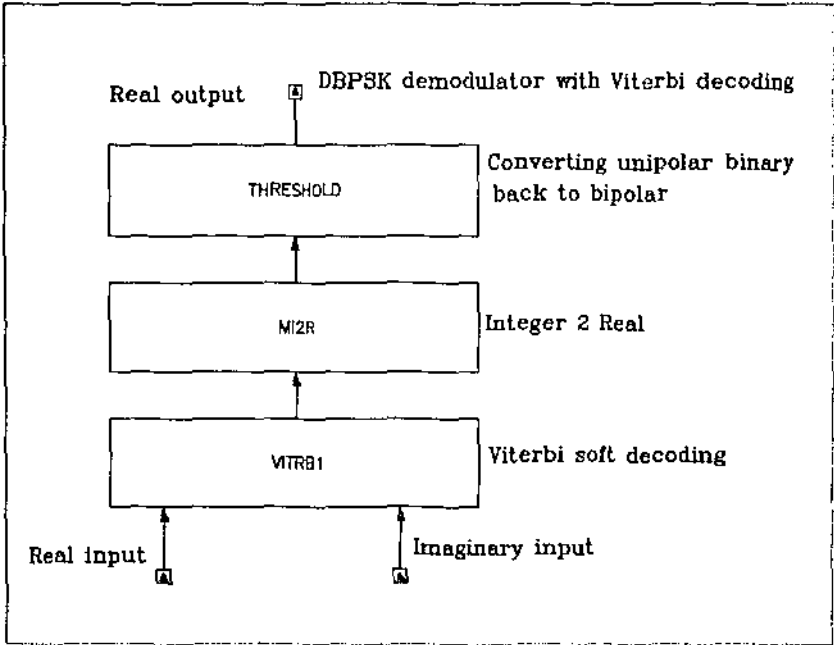


Figure 41: COSSAP model (DBPSK_V)

| |
|--|
| .NUMBER_OF_STATES |
| 2 |
| .MAXIMUM_#_OF_PARALLEL_BRANCHES_P |
| 1 |
| .NUMBER_OF_BRANCHES_JOINING_AT_ONE_STATE |
| 2 |
| .NUMBER_M_OF_CHANNEL_SYMBOLS |
| 2 |
| .SELECT_MODULATION_(MPSK=0_QAM=1) |
| 0 |
| .WITH_DIFFERENTIAL_CODING_(YES=1_NO=0) |
| 1 |
| *.TRELLIS_TABLE_(STATE,CHANNEL_SYMBOL)" |
| ----- ----- ----- ----- ----- ----- |
| 1 1 0 |
| 1 2 1 |
| 2 1 1 |
| 2 2 0 |
| .COMPONENT_TABLE_FOR_CHANNEL_SYMBOLS |
| ----- ----- ----- ----- |
| 0 1 0 |
| 1 -1 0 |
| .END |

Table 26: Trellis/Viterbi encoding/decoding parameter file for COSSAP model *DBPSK_T* & *DBPSK_V*

C.5 QPSK Modulator (*QPSK_T*)

| |
|---------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| QPSK_T |
| |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |

Table 27: Configuration file for COSSAP model *QPSK_T*

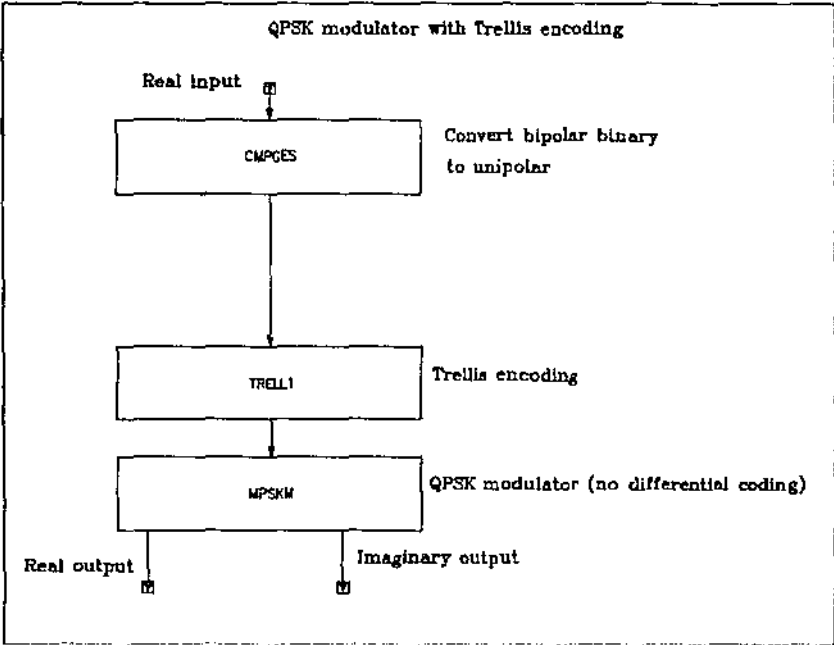


Figure 42: COSSAP model (*QPSK_T*)

C.6 QPSK Demodulator (QPSK_V)

| |
|--|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| MODEL NAME : |
| QPSK_V |
| PARAMETERS : |
| (I) \$SURVIVOR_DEPTH |
| used in block VITRB1 (4) : |
| SURVIVOR_DEPTH..... = \$SURVIVOR_DEPTH |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| (R) \$INPUT_IMAG |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |

Table 28: Configuration file for COSSAP model QPSK_V

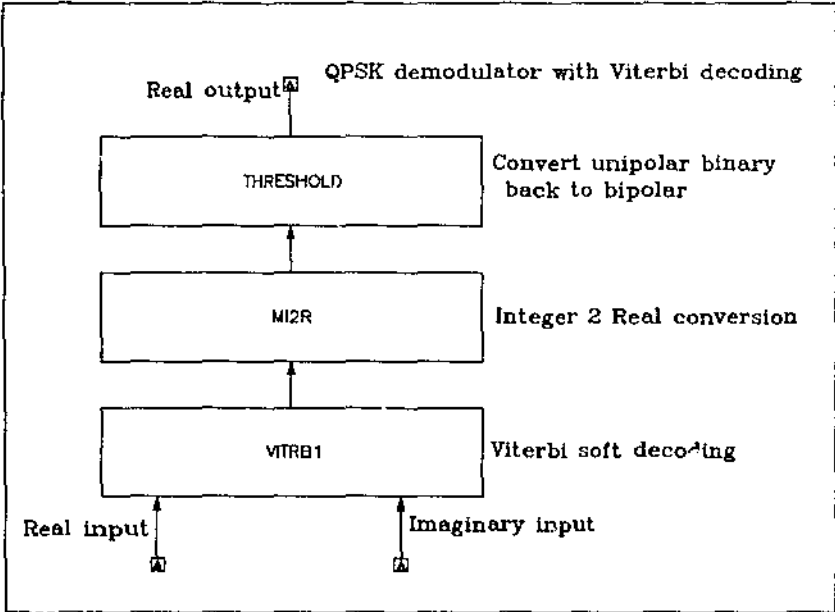


Figure 43: COSSAP model (QPSK_V)

| |
|--|
| .NUMBER_OF_STATES |
| 4 |
| .MAXIMUM_#_OF_PARALLEL_BRANCHES_P |
| 1 |
| .NUMBER_OF_BRANCHES_JOINING_AT_ONE_STATE |
| 2 |
| .NUMBER_M_OF_CHANNEL_SYMBOLS |
| 4 |
| .SELECT_MODULATION_(MPSK=0_QAM=1) |
| 0 |
| .WITH_DIFFERENTIAL_CODING_(YES=1_NO=0) |
| 0 |
| *.TRELLIS_TABLE_(STATE,CHANNEL_SYMBOL)" |
| ---- ---- ---- ---- ---- ---- |
| 1 1 0 |
| 1 3 2 |
| 2 1 2 |
| 2 3 0 |
| 3 2 1 |
| 3 4 3 |
| 4 2 3 |
| 4 4 1 |
| .COMPONENT_TABLE_FOR_CHANNEL_SYMBOLS |
| ----- ----- ----- ----- |
| 0 1 0 |
| 1 0 1 |
| 2 -1 0 |
| 3 0 -1 |
| .END |

Table 29: Trellis/Viterbi encoding/decoding parameter file for COSSAP model *QPSK_T* & *QPSK_V*

C.7 DQPSK Modulator (DQPSK_T)

| |
|---------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| DQPSK_T |
| |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |

Table 30: Configuration file for COSSAP model DQPSK_T

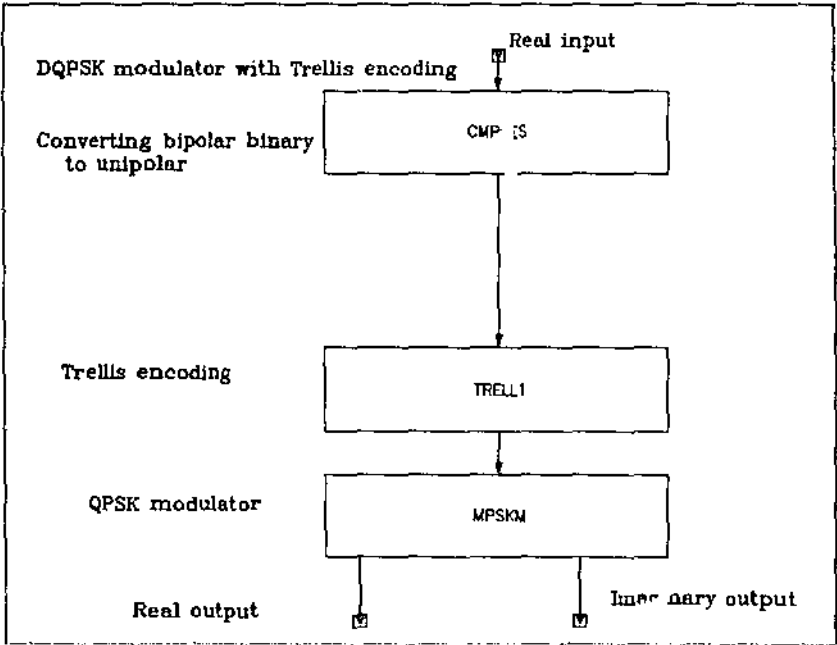


Figure 44: COSSAP model (DQPSK_T)

C.8 DQPSK Demodulator (DQPSK_V)

| |
|--|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| MODEL NAME : |
| DQPSK_V |
| PARAMETERS : |
| (I) \$SURVIVOR_DEPTH |
| used in block VITRB1 (4) : |
| SURVIVOR_DEPTH..... = \$SURVIVOR_DEPTH |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| (R) \$INPUT_IMAG |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |

Table 31: Configuration file for COSSAP model DQPSK_V

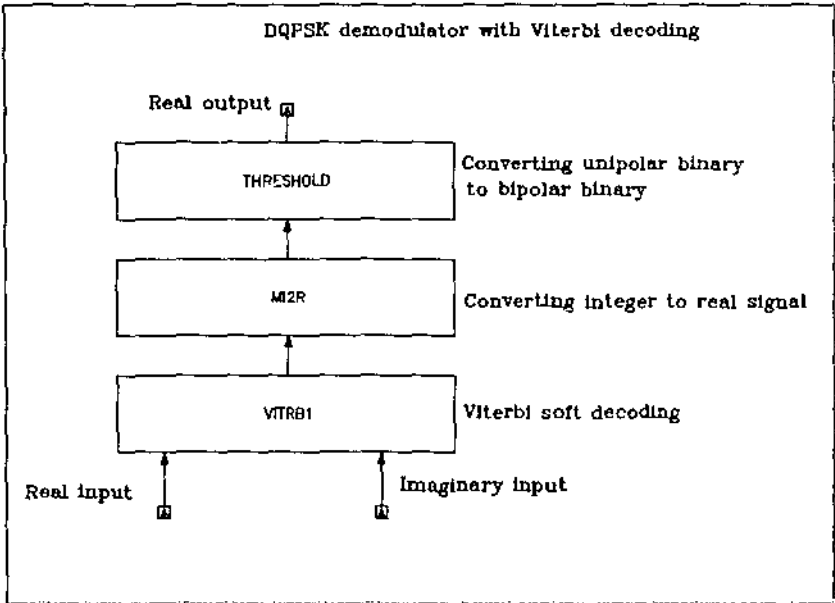


Figure 45: COSSAP model (DQPSK_V)

| |
|--|
| .NUMBER_OF_STATES |
| 4 |
| .MAXIMUM_#_OF_PARALLEL_BRANCHES_P |
| 1 |
| .NUMBER_OF_BRANCHES_JOINING_AT_ONE_STATE |
| 2 |
| .NUMBER_M_OF_CHANNEL_SYMBOLS |
| 4 |
| .SELECT_MODULATION_(MPSK=0_QAM=1) |
| 0 |
| .WITH_DIFFERENTIAL_CODING_(YES=1_NO=0) |
| 1 |
| *.TRELLIS_TABLE_(STATE,CHANNEL_SYMBOL)" |
| ----- ----- ----- ----- ----- ----- |
| 1 1 0 |
| 1 3 2 |
| 2 1 2 |
| 2 3 0 |
| 3 2 1 |
| 3 4 3 |
| 4 2 3 |
| 4 4 1 |
| .COMPONENT_TABLE_FOR_CHANNEL_SYMBOLS |
| ----- ----- ----- ----- |
| 0 1 0 |
| 1 0 1 |
| 2 -1 0 |
| 3 0 -1 |
| .END |

Table 32: Trellis/Viterbi encoding/decoding parameter file for COSSAP model *DQPSK_T* & *DQPSK_V*

Appendix D: CDMA multiplexer and demultiplexer blocks

Following are the configuration files and COSSAP models for the 16-bits Rademacher-Walsh Orthogonal Sequence Multiplexer Functional Block (*Walsh_Mux*) and the 16-bits Rademacher-Walsh Orthogonal Sequence Demultiplexer Functional Block (*Walsh_DMux*). Their respective complex signals counterpart is also illustrated.

D.1 16-bits Rademacher-Walsh Orthogonal Sequence Multiplexer Functional Block (*Walsh_Mux*)

| |
|-------------------------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| Walsh_Mux |
| |
| PARAMETERS : |
| (I) \$Walsh_Num_Bits |
| used in block HOLDR (4) : |
| HOLD_FACTOR..... = \$Walsh_Num_Bits |
| |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| |
| INPUT_DATASETS : |
| \$Walsh |
| used in block Walsh_Gen (005) |
| dataset name : PATTERN |
| dataset number : 1 |

Table 33: Configuration file for COSSAP model *Walsh_Mux*

Walsh Multiplexer

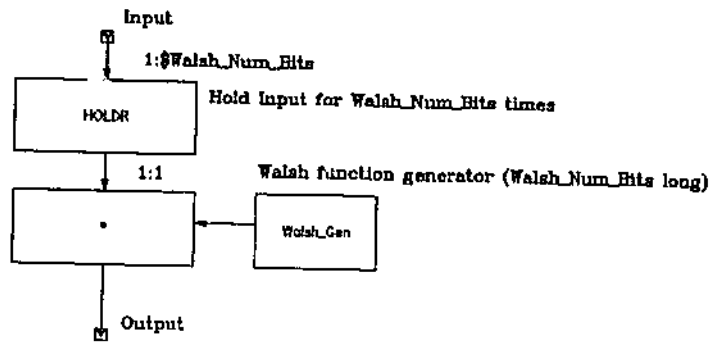


Figure 46: COSSAP model (*Walsh_Mux*)

D.1.1 16-bits Rademacher-Walsh Orthogonal Sequence Multiplexer Functional Block
(Walsh_Mux_QC) - Complex Signals

| |
|---|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| MODEL NAME : |
| Walsh_Mux_QC |
| PARAMETERS : |
| (I) \$Walsh_Num_Bits |
| used in block Walsh_Mux (6) : |
| Walsh_Num_Bits..... = \$V:alsh_Num_Bits |
| used in block Walsh_Mux (5) : |
| Walsh_Num_Bits..... = \$Walsh_Num_Bits |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| (R) \$INPUT_IMAG |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |
| INPUT_DATASETS : |
| \$Walsh |
| used in block Walsh_Mux (005) |
| dataset name : Walsh |
| dataset number : 1 |
| used in block Walsh_Mux (006) |
| dataset name : Walsh |
| dataset number : 1 |

Table 34: Configuration file for COSSAP model *Walsh_Mux_QC*

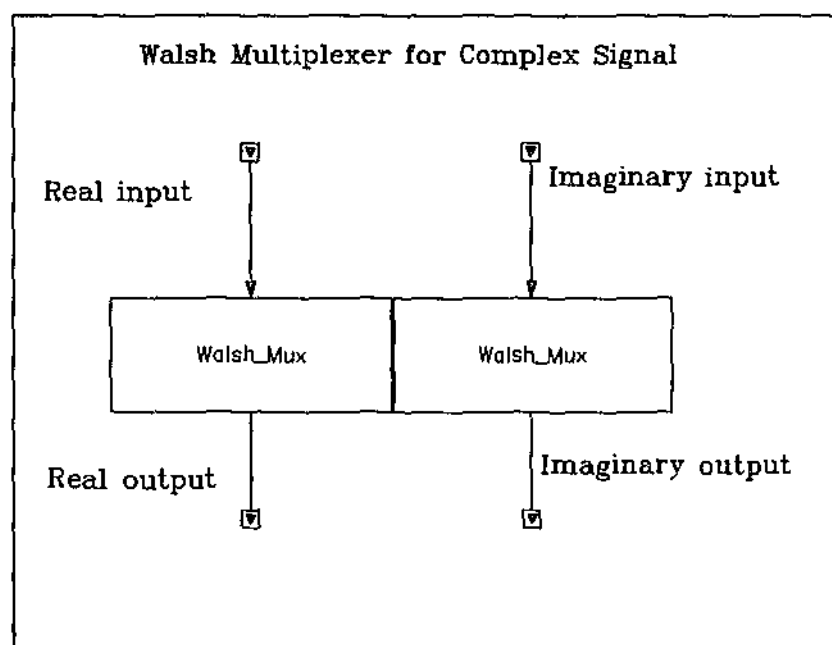


Figure 47: COSSAP model (*Walsh_Mux_QC*)

D.2 16-bits Rademacher-Walsh Orthogonal Sequence Demultiplexer Functional Block (*Walsh_DMux*)

| |
|---------------------------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| MODEL NAME : |
| Walsh_DMux |
| PARAMETERS : |
| (I) \$Walsh_Num_Bits |
| used in block Integrator (7) : |
| NumberOfItems..... = \$Walsh_Num_Bits |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| INPUT_DATASETS : |
| \$Walsh |
| used in block Walsh_Gen (005) |
| dataset name : PATTERN |
| dataset number : 1 |

Table 35: Configuration file for COSSAP model *Walsh_DMux*

Walsh Demultiplexer

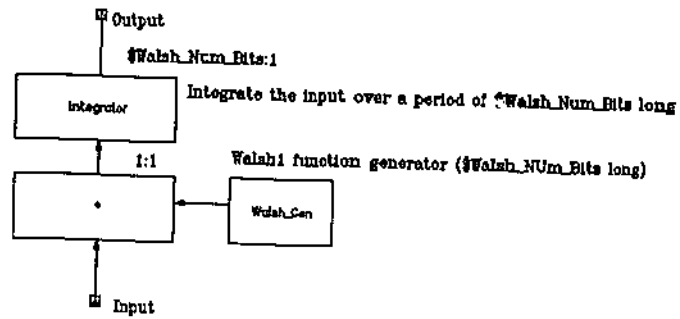


Figure 48: COSSAP model (*Walsh_DMux*)

D.2.1 16-bits Rademacher-Walsh Orthogonal Sequence Demultiplexer Functional Block (Walsh_DMux_QC) - Complex Signals

| |
|--|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| MODEL NAME : |
| Walsh_DMux_QC |
| PARAMETERS : |
| (I) \$Walsh_Num_Bits |
| used in block Walsh_DMux (5) : |
| Walsh_Num_Bits..... = \$Walsh_Num_Bits |
| used in block Walsh_DMux (4) : |
| Walsh_Num_Bits..... = \$Walsh_Num_Bits |
| INPUT_PORTS : |
| (R) \$INPUT_REAL |
| (R) \$INPUT_IMAG |
| OUTPUT_PORTS : |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |
| INPUT_DATASETS : |
| \$Walsh_d |
| used in block Walsh_DMux (004) |
| dataset name : Walsh |
| dataset number : 1 |
| used in block Walsh_DMux (005) |
| dataset name : Walsh |
| dataset number : 1 |

Table 36: Configuration file for COSSAP model *Walsh_DMux_QC*

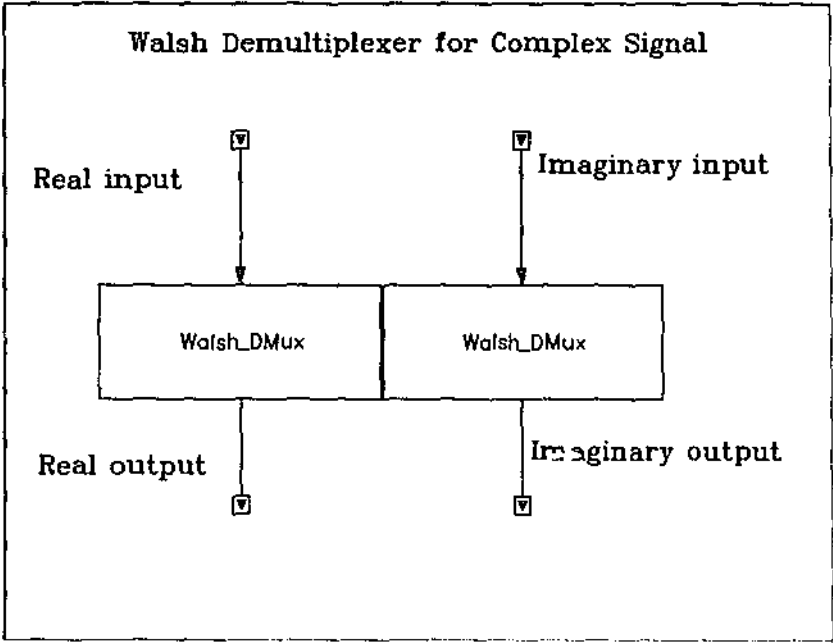


Figure 49: COSSAP model (*Walsh_DMux_QC*)

| # | 16 (R) |
|---|--------|
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | 1 |
| | -1 |
| | -1 |
| | -1 |
| | -1 |
| | -1 |
| | -1 |
| | -1 |
| | -1 |
| | -1 |
| # | -1 (R) |

Table 37: Sample of 16-bits Rademacher-Walsh sequence input data file

Appendix E: Channel Modelling

Following are the configuration files and COSSAP models for the **Additive White Gaussian Noise (AWGNQC)**, **Rayleigh Distribution Multipath Fading (M_{RAYLEIGH})** and **Rician Distribution Multipath Fading ($RICE$)** channels. All these channel models are from the COSSAP system libraries and each supports the use of complex signals.

E.1 Additive White Gaussian Noise (AWGNQC) Channel

| | |
|-----------------------------------|--|
| ***** | |
| block: awgnqc BLOCK CONFIGURATION | |
| ***** | |
| BLOCK : | |
| model name | : awgnqc |
| model id | : 153 |
| LIBRARY : | |
| name | : mod63.mdb |
| directory | : \${COSSAP_MODELS}/mod63 |
| type | : database |
| installation id | : 0 (all machines at Synopsys DSP Tools, Aachen) |
| user id | : 0 (SYNOPTSYS) |
| library id | : 0 |

Table 38: Configuration file for COSSAP model AWGNQC

| |
|---|
| sequence number : 4 |
| PARAMETERS : |
| 1 (R) SNR..... \$SNR |
| 2 (R) SIGNAL_POWER..... \$Signal_Power |
| 3 (R) FACTOR..... 1.0 |
| 4 (I) SIGNAL_POWER_KNOWN..... 1 |
| 5 (I) RANDOM_SEED..... 24091971 |
| INPUT_PORTS : |
| 1 (R) IN_REAL..... not connected |
| 2 (R) IN_IMAG..... not connected |
| OUTPUT_PORTS : |
| 1 (R) OUT_REAL..... not connected |
| 2 (R) OUT_IMAG..... not connected |
| SELECTED IMPLEMENTATION : |
| implementation type..... = DEFAULT_IMPLEMENTATION |

Table 38: Configuration file for COSSAP model *AWGNQC*

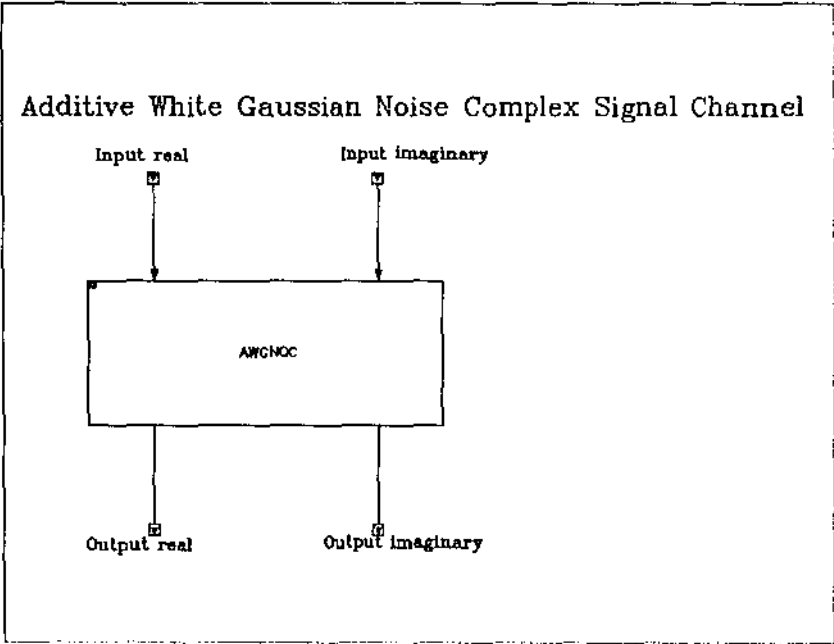


Figure 50: COSSAP model (*AWGNQC*)

E.2 Rayleigh Distribution Multipath Fading (*M_RAYLEIGH*) Channel

| |
|--|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| M_RAYLEIGH |
| |
| PARAMETERS : |
| (I) \$VELOCITY_KM_H |
| used in block: LININT (8) : |
| INTERPOLATION_RATIO..... = |
| INT(2928/REAL(\$VELOCITY_KM_H)+0.5) |
| (R) \$POWER_DB |
| used in block MCVCSM (15) : |
| SCALAR_REAL..... = 10**(\$POWER_DB/20) |
| (I) \$SEED |
| used in block WGNQC (4) : |
| RANDOM_SEED..... = \$SEED |
| |
| INPUT_PORTS : |
| (R) \$IN_REAL |
| (R) \$IN_IMAG |
| |
| OUTPUT_PORTS : |
| (R) \$OUT_REAL |
| (R) \$OUT_IMAG |

Table 39: Configuration file for COSSAP model *M_RAYLEIGH*

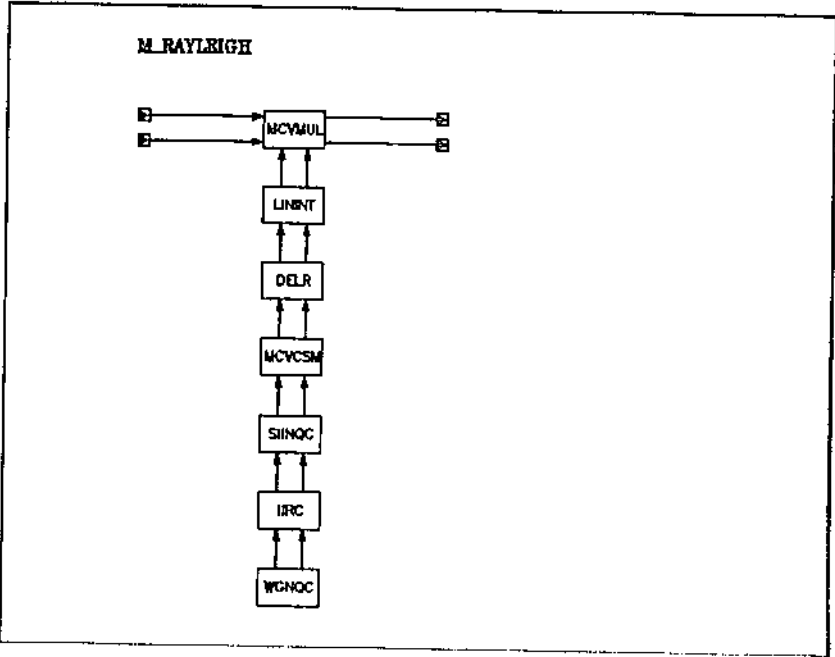


Figure 51: COSSAP model (*M_RAYLEIGH*)

E.3 Rician Distribution Multipath Fading (*RICE*) Channel

| |
|--|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME : |
| RICE |
| |
| PARAMETERS : |
| (I) \$VELOCITY_KM_H |
| used in block FDEVC (13) : |
| LAMBDA..... = \$VELOCITY_KM_H*5.38125E-7 |
| used in block M_RAYLEIGH (5) : |
| VELOCITY_KM_H..... = \$VELOCITY_KM_H |
| (R) \$TIME_U_SEC |
| used in block DELAYR (6) : |
| DELAY..... = INT(10*\$TIME_U_SEC) |
| (I) \$SEED |
| used in block M_RAYLEIGH (5) : |
| SEED..... = \$SEED |
| |
| INPUT_PORTS : |
| (R) \$IN_REAL |
| (R) \$IN_IMAG |
| |
| OUTPUT_PORTS : |
| (R) \$OUT_REAL |
| (R) \$OUT_IMAG |

Table 40: Configuration file for COSSAP model *RICE*

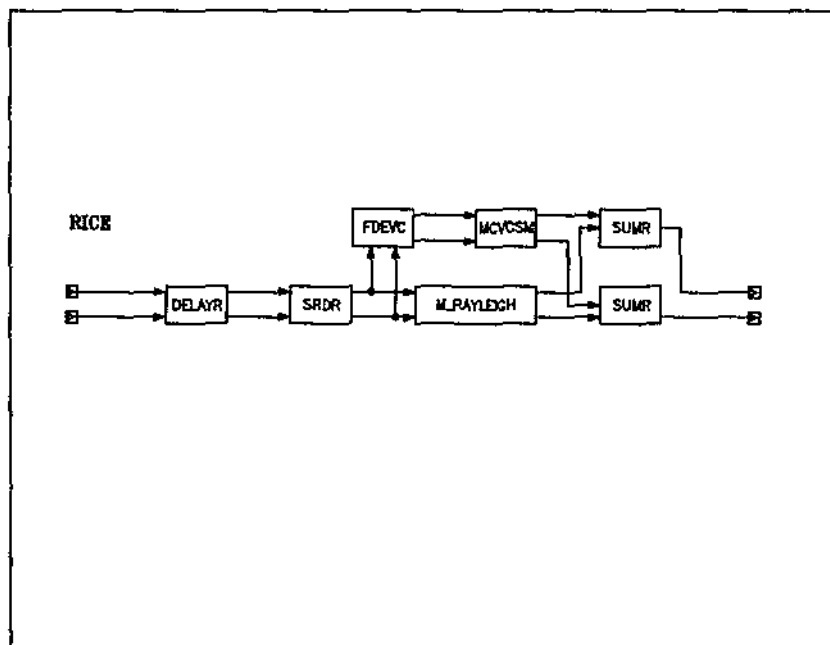


Figure 52: COSSAP model (*RICE*)

Appendix F: Arithmetic signals adder blocks

Following are the configuration files and COSSAP models for the Arithmetic Adder for 9 Complex Signals (*ADD9_QC*) and the Arithmetic Adder for 16 Complex Signals (*ADD16_QC*).

F.1 Arithmetic Adder for 9 Complex Signals (*ADD9_QC*)

| |
|---------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME: |
| ADD9_QC |
| |
| INPUT_PORTS: |
| (R) \$INPUT_REAL1 |
| (R) \$INPUT_IMAG1 |

Table 41: Configuration file for COSSAP model *ADD9_QC*

| |
|-------------------|
| (R) \$INPUT_REAL2 |
| (R) \$INPUT_IMAG2 |
| (R) \$INPUT_REAL3 |
| (R) \$INPUT_IMAG3 |
| (R) \$INPUT_REAL4 |
| (R) \$INPUT_IMAG4 |
| (R) \$INPUT_REAL5 |
| (R) \$INPUT_IMAG5 |
| (R) \$INPUT_REAL6 |
| (R) \$INPUT_IMAG6 |
| (R) \$INPUT_REAL7 |
| (R) \$INPUT_IMAG7 |
| (R) \$INPUT_REAL8 |
| (R) \$INPUT_IMAG8 |
| (R) \$INPUT_REAL9 |
| (R) \$INPUT_IMAG9 |
| |
| OUTPUT_PORTS: |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |

Table 41: Configuration file for COSSAP model *ADD9_QC*

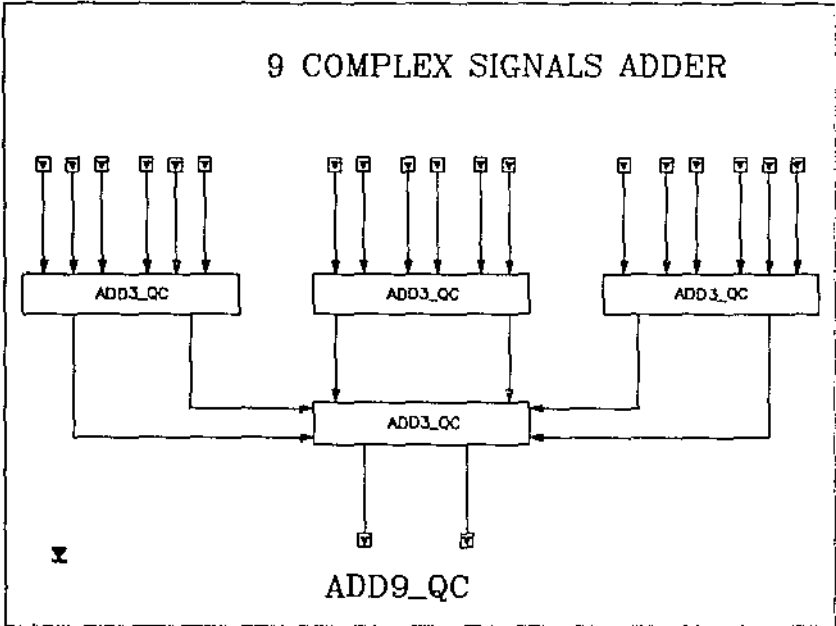


Figure 53: COSSAP model (*ADD9_QC*)

F.2 Arithmetic Adder for 16 Complex Signals (*ADD16_QC*)

| |
|---------------------|
| ***** |
| MODEL CONFIGURATION |
| ***** |
| |
| MODEL NAME: |
| ADD16_QC |
| |
| INPUT_PORTS: |
| (R) \$INPUT_REAL1 |
| (R) \$INPUT_IMAG1 |
| (R) \$INPUT_REAL2 |
| (R) \$INPUT_IMAG2 |
| (R) \$INPUT_REAL3 |
| (R) \$INPUT_IMAG3 |
| (R) \$INPUT_REAL4 |
| (R) \$INPUT_IMAG4 |
| (R) \$INPUT_REAL5 |
| (R) \$INPUT_IMAG5 |
| (R) \$INPUT_REAL6 |
| (R) \$INPUT_IMAG6 |
| (R) \$INPUT_REAL7 |
| (R) \$INPUT_IMAG7 |
| (R) \$INPUT_REAL8 |
| (R) \$INPUT_IMAG8 |

Table 42: Configuration file for COSSAP model *ADD16_QC*

| |
|--------------------|
| (R) \$INPUT_REAL9 |
| (R) \$INPUT_IMAG9 |
| (R) \$INPUT_REAL10 |
| (R) \$INPUT_IMAG10 |
| (R) \$INPUT_REAL11 |
| (R) \$INPUT_IMAG11 |
| (R) \$INPUT_REAL12 |
| (R) \$INPUT_IMAG12 |
| (R) \$INPUT_REAL13 |
| (R) \$INPUT_IMAG13 |
| (R) \$INPUT_REAL14 |
| (R) \$INPUT_IMAG14 |
| (R) \$INPUT_REAL15 |
| (R) \$INPUT_IMAG15 |
| (R) \$INPUT_REAL16 |
| (R) \$INPUT_IMAG16 |
| |
| OUTPUT_PORTS: |
| (R) \$OUTPUT_REAL |
| (R) \$OUTPUT_IMAG |

Table 42: Configuration file for COSSAP model *ADD16_QC*

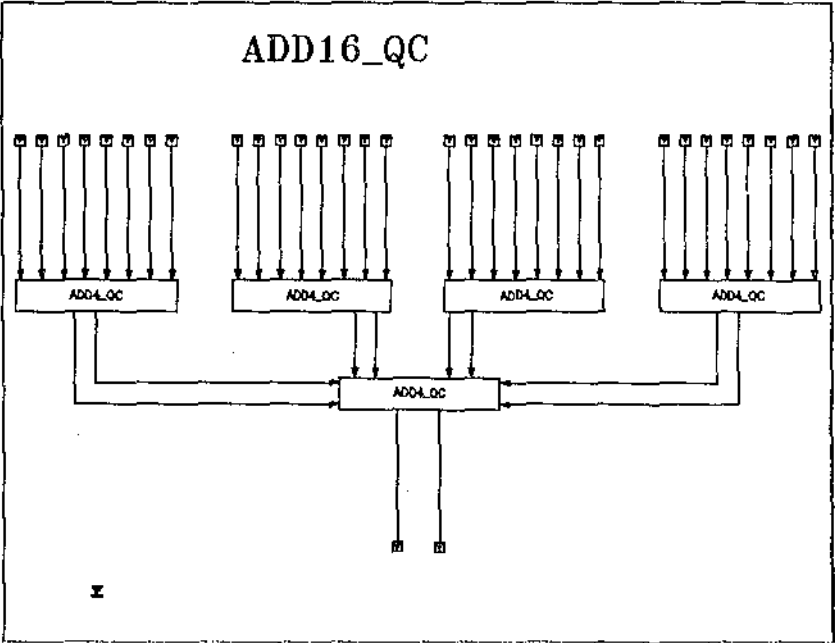


Figure 54: COSSAP model (*ADD16_QC*)